University of Alberta

REAL-TIME HARDWARE-IN-THE-LOOP SIMULATION AND CONTROL OF VSC-HVDC SYSTEM

by

Md Omar Faruque ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**

Department of Electrical and Computer Engineering

Edmonton, Alberta
Spring 2008

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-45424-4
Our file Notre référence
ISBN: 978-0-494-45424-4

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve [illegible] propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

New power electronic technologies such as the Voltage Source Converter (VSC) based High Voltage Direct Current (HVDC) transmission system and Flexible Alternating Current Transmission Systems (FACTS) are finding increasing application in modern powers systems. Before applying these technologies and their controllers in the real field, it is necessary to study their behavior and interaction with the system using comprehensive simulation tools. Real-time Hardware-In-the-Loop (HIL) simulation offers an accurate replica of the system when testing new controllers/devices/relays under various conditions with the aid of advanced state-of-the-art digital technology.

To realize an accurate HIL simulation, numerous challenges needed to be overcome such as accurate modeling of both electrical and power electronic systems, choice of proper discretization methods for circuit elements, synchronization of discrete switching events and interfacing of the real controller hardware with the digital simulator. This thesis addresses these problems and proposes solutions to overcome them. Suitable discretization methods and several algorithms for accounting switching events in fixed-step digital simulation are proposed. A detailed mathematical model for the VSC-HVDC system is developed and implemented as an off-line simulation program using the proposed algorithms. A digital control scheme for the operation of the VSC-HVDC is designed, developed and implemented. Two well-known transient simulation programs, PSCAD/EMTDC and MATLAB/SIMULINK are used to validate the control design through a comparative study of the simulation results. Next, a real-time HIL simulation is implemented for the VSC-HVDC system and interfaced with the digital controller. A computing node of a PC-cluster is used as a simulator and another node is used as the digital controller. The simulator and the controller are connected through I/O ports which facilitate the exchange of analog and digital signals. Finally, a 4kW VSC-HVDC experimental setup was developed where the same digital controller as in the HIL simulation supplies the necessary gating pulses for the VSCs. A comparative study of the results obtained through off-line, HIL simulation and experiment shows excellent agreement.

Acknowledgements

I would like to express my sincere gratitude and deep appreciation to my supervisor, Dr. Venkata Dinavahi for his support, guidance and encouragement while I pursued this research at the University of Alberta. His insightful guidance, passion and enthusiasm for the research was an invaluable motivation for me.

It was a pleasure to work with my friends and colleagues in the RTX-Lab and I would like to thank them for their helpful advice and selfless assistance. Special thanks go to Lok-Fu (Paul), Xin and Sami whose companionship has made these long years of research joyful and exciting. Many thanks go to Albert for his technical assistances in the lab.

I wish to extend my deepest gratitude toward my parents for their love, caring, and endless sacrifices throughout my life. Also thanks to my parents-in-law whose inspiration and support was instrumental for me to complete this work.

Finally, my sincere thanks and appreciation to my wife, Zita for her support, sacrifice and encouragement. I am indebted to her and to my son, Zibran who missed my care and love during these years.

Financial help from the University of Alberta, NSERC, Killam Memorial Trusts and iCORE for this research work is greatly appreciated.

Contents

1 Introduction 1

1.1 Motivation for this Work 2
1.2 Real-Time Hardware-In-the-Loop (HIL) Simulation 5
1.3 Literature Review 9
    1.3.1 Modeling and Solution Techniques 9
    1.3.2 Switching Synchronization 12
    1.3.3 Off-Line Electromagnetic Transient Simulation 13
    1.3.4 Evolution of Real-Time Digital Simulator 15
1.4 Thesis Objectives 16
1.5 Thesis Outline 17

2 Network Discretization and Solution Techniques 19

2.1 Introduction 19
2.2 Step-Invariant and Ramp-Invariant Transformation 20
    2.2.1 Transfer Function Approach 21
    2.2.2 State-Space Approach 25
2.3 Accuracy of SIT and RIT 29
    2.3.1 Errors in Transfer Function Approach 29
    2.3.2 Errors in State-Space Approach 30
2.4 Stability of SIT and RIT Algorithms 32
2.5 Illustrative Examples 34
    2.5.1 Transfer Function Approach 34
        2.5.1.1 Comparison of Accuracy of SIT, TSSIT and RIT . . 36
    2.5.2 State-Space Approach 44

2.6 Summary 50

3 **Switching Events Synchronization** **52**

3.1 Introduction 52
3.2 Multiple Inter-step Switchings 54
3.3 Algorithms for the accounting of multiple switches 56
3.3.1 Fixed Time-Grid Algorithms 58
3.3.2 Variable Time-Grid Algorithms 63
3.4 Case Study: PWM VSC System 66
3.4.1 System Model 66
3.4.2 Digital Controller 67
3.5 Results and Discussion 68
3.5.1 Open Loop Control 68
3.5.2 Closed-Loop Control 75
3.5.3 Execution Time 76
3.6 Summary 77

4 **Modeling, Control and Off-line Simulation of VSC-HVDC** **79**

4.1 Introduction 79
4.2 VSC-HVDC Test System 80
4.2.1 Operation of VSC-HVDC 81
4.3 Control of VSC-HVDC 83
4.3.1 Control of Converter 1 83
4.3.1.1 PWM and Gate Pulse Generation 86
4.3.1.2 Control Scheme in d-q Frame 87
4.3.1.3 Design of PI controllers 91
4.3.2 Control of Converter 2 93
4.4 Modeling of VSC-HVDC for Off-line Simulation 95
4.4.1 Modeling of VSC-HVDC System in PSCAD/EMTDC 95
4.4.2 Modeling of VSC-HVDC System in MATLAB/SIMULINK . . 97
4.5 Off-line Simulation Results 98
4.5.1 Steady-State 99

4.5.2 Transients 104
4.6 Execution Time and Memory 115
4.7 General Remarks 115
4.8 Summary 116

5 Real-Time HIL Simulation of VSC-HVDC and Experimental Validation 118
5.1 Introduction 118
5.2 Modeling of VSC-HVDC for Real-Time Simulation 119
5.2.1 AC System 119
5.2.2 Converters 120
5.2.3 DC-Link 122
5.2.4 Controller 122
5.2.5 State-Space Model of the Overall System 123
5.2.6 Discrete-Time Model of the Overall System 125
5.3 Off-line Simulation using C 127
5.4 Real-Time Digital Simulator 128
5.5 HIL Simulation of VSC-HVDC 130
5.5.1 Hardware Setup 131
5.6 Implementation of the HIL Simulation 133
5.6.1 Digital Controller 135
5.6.2 PWM Generation Inside FPGA 137
5.6.3 Modeling the Electrical System 138
5.6.4 Capturing of Switching Events 139
5.6.5 Building of the S-function 142
5.7 Experimental Setup 143
5.8 Results and Discussion 148
5.8.1 Steady-State 148
5.8.2 Effect of Distorted Supply Voltage on Steady-State Results . . 157
5.8.3 Transients 159
5.9 Execution Time for HIL Simulation 167

5.10 Summary 168

6 Conclusions **170**

6.1 Future Work 173

Bibliography **175**

Appendix A **184**

A.1 SIT and RIT for R, L, C, RL, RC, LC and RLC Network 184

A.1.1 Resistive Network 184

A.1.2 Inductive Network 185

A.1.3 Capacitive Network 186

A.1.4 RL Network 186

A.1.5 LC Network 188

A.1.6 RC Network 188

A.1.7 RLC Network 189

Appendix B **192**

B.1 Simulation Diagrams of VSC-HVDC 192

Appendix C **198**

C.1 Control of VSC-HVDC using SIMULINK and RTLAB Toolbox . . . 198

List of Tables

2.1 Norton equivalents for circuit elements using SIT. 37
2.2 Norton equivalents for circuit elements using TSSIT. 37
2.3 Norton equivalents for circuit elements using RIT. 38
2.4 Discretization errors produced by different methods for $\triangle t = \tau = 50\mu s$. 39
2.5 Discretization errors produced by different methods for $\triangle t = \tau = 50\mu s$.(contd.) 41
2.6 Discretization errors produced by different methods for $\triangle t = 5\tau = 250\mu s$. 41
2.7 Discretization errors produced by different methods for $\triangle t = 5\tau = 250\mu s$.(contd.) 41
2.8 Discretization errors produced by different methods for sinusoidal input at $\triangle t = 10\mu s$. 43
2.9 Discretization errors produced by different methods for sinusoidal input solved using state-space method at a time-step of $\triangle t = 50\mu$ s 48
2.10 Discretization errors produced by different methods for sinusoidal input solved using state-space method at a time-step of $\triangle t = 50\mu$ s. (contd.) 48

3.1 Variation of switching events with simulation step-size for a total simulation period of 2 s with carrier frequency of 1 kHz. 69
3.2 Fundamental error ε and THD of I_a for varying $\triangle t$. 71
3.3 Comparison of MAD and SNR (dB) of i_q produced by different algorithms for $\triangle$t=100 μs 76

4.1 IGBT parameters used for the simulation. 96

5.1 Harmonic components of laboratory supply voltage. 159
5.2 Execution times for individual tasks within one time-step of $10\mu s$. . . 168

List of Figures

1.1 (a) Real-Time and (b) Non-Real-Time or Off-Line Simulation. 6

1.2 HIL simulation (a) Control hardware-in-the-loop (b) Power hardware-in-the-loop. 7

2.1 (a) Step-Invariant input (b) Ramp-Invariant input. 21

2.2 (a) Analog plant (b) Required digital plant for simulation (c) Conversion to digital plant (d) Converted digital plant 22

2.3 (a) Piecewise-constant integrand (b) Piecewise-linear integrand. . . . 28

2.4 Difference between local and total errors. 30

2.5 An RL network excited by a step-function. 35

2.6 Norton Equivalent of RL network (a) Trapezoidal rule (b) TSSIT. . . 36

2.7 Response of the RL network excited by a step-function, using different discretization methods with a time-step $\Delta t = \tau = 50\mu s$. 40

2.8 Response of the RL network, excited by a step-function, using different discretization methods with a time-step $\Delta t = 5\tau = 250\mu s$. 42

2.9 Steady-state response of the RL network when excited by a sinusoidal voltage source and simulated with a time-step $\Delta t = 10\mu s$. 43

2.10 Transient response of the RL network when excited by a sinusoidal voltage source and simulated with a time-step $\Delta t = 10\mu s$. 44

2.11 (a) RLC network, (b) step-function input, (c) Equivalent state-space model 47

2.12 (a) Step response of an RLC network solved by state-space solution technique with different discretization methods at a time-step $\Delta t = 50\mu s$, (b) Zoomed view of the response. 49

2.13 (a) Step response of an RLC network solved by state-space solution technique with different discretization methods at a time-step $\Delta t = 50\mu s$, (b) Zoomed view of the response. 49

2.14 (a) Step response of an RLC network solved by state-space solution technique using different discretization methods with a time-step $\Delta t = 250\mu s$, (b) Zoomed view of the response. 50

2.15 (a) Step response of an RLC network solved by state-space solution technique using different discretization methods with a time-step $\Delta t = 250\mu s$, (b) Zoomed view of the response. 50

3.1 Current chopping of a diode 53

3.2 Multiple inter-step switching events in digital simulation. 55

3.3 Digital simulation algorithms for multiple switching events based on a fixed time-grid. ⌒⌒ : Full-state calculation, ⊙ : Linear Interpolation, → : Linear Extrapolation. 57

3.4 Variable time-grid Algorithm VIII for a two switching events. ⌒ : Full-state calculation, • : Switching update, → : Linear extrapolation. 64

3.5 PWM VSC system and its digital controller 66

3.6 Illustration of multiple switchings through a detailed view of the simulation for one complete cycle of control signal suing D-STATCOM test system. 69

3.7 Detailed view of the steady state current I_a. Curve 1: Fixed time-step algorithm with $\Delta t = 10\mu s$,Curve 2: Fixed time-step algorithm with $\Delta t = 150\mu s$, Curve 3: Algorithm VII with $\Delta t = 150\mu s$. 72

3.8 Frequency spectrum of I_a. Data 1: Fixed time-step algorithm with $\Delta t = 10\mu s$, Data 2: Algorithm VII with $\Delta t = 150\mu s$, Data 3: Fixed time-step algorithm with $\Delta t = 150\mu s$. 73

3.9 Plot of percentage fundamental error ε with respect to simulation time-step 74

3.10 Plot of THD with respect to simulation time-step 74

3.11 Step response in i_q simulated using the (A) Fixed time-step algorithm with $\Delta t = 10\mu s$ (B) Fixed time-step algorithm with $\Delta t = 100\mu s$, (C) Algorithm VII with $\Delta t = 100\mu s$ and (D) Algorithm VII with $\Delta t = 100\mu s$ but $\mathbf{x_2}$ corrected. 75
3.12 Execution time for the time-step involving multiple switching events using various algorithms with Δt=100 μs 78

4.1 Schematic diagram of VSC-HVDC test system. 81
4.2 (a) Phasor diagram indicating the changes in power flow due to the changes in the magnitude and angle of $V2i$, (b) Locus of maximum P_2 and Q_2 obtained through the variation of $V2i$. 82
4.3 Overall control scheme of the VSC-HVDC (a) Control of converter 1 (b) Control of converter 2 84
4.4 Vector diagram of coordinate transformation system. 85
4.5 Sinusoidal PWM: (a) Comparison of modulating signal and the carrier, (b) Generated pulse for one switching device. 87
4.6 Decoupled control scheme for converter 1. 90
4.7 Control loops (a) Inner loop PI controller for i_d and i_q (b) Outer loop PI controller for V_{dc}. 91
4.8 PI controller design: (a) Root locus including boundary limits (b) Zoomed view of the selected gain point. 93
4.9 PI controller design for DC-link: (a) Root locus plot (b) Step response. 94
4.10 Control scheme for converter 2. 94
4.11 Voltages and currents at converter 1 using both PSCAD/EMTDC and PSB/SIMULINK. 100
4.12 Voltages and currents at converter 2 using both PSCAD/EMTDC and PSB/SIMULINK. 101
4.13 DC voltages and powers using both PSCAD/EMTDC and PSB/SIMULINK. 102
4.14 Voltages and currents at converter 1 using both PSCAD/EMTDC and PSB/SIMULINK. 104

4.15 Voltages and currents at converter 2 using both PSCAD/EMTDC and PSB/SIMULINK. 105
4.16 DC voltages and powers using both PSCAD/EMTDC and PSB/SIMULINK. 106
4.17 DC voltages and powers using both PSCAD/EMTDC and PSB/SIMULINK. 107
4.18 Responses of dc-link voltage and power quantities due to a short circuit at dc-link 109
4.19 Waveforms of voltages and currents while there is a short circuit at dc-link 110
4.20 Reference values for power and dc-link voltage. 111
4.21 Transients in various power levels. 112
4.22 Responses of DC-link voltage and powers due to the loss of grid supply to converter 2. 113
4.23 Voltages and currents while converter 2 lost its supply from 0.5s to 0.6s . 114

5.1 Equivalent circuit diagram of the VSC-HVDC test system. 120
5.2 Switching function model of voltage source converter. 122
5.3 Snapshot of the PC-Cluster based real-time digital simulator installed in the RTX-Lab. 129
5.4 Hardware architecture of the simulator installed in the RTX-Lab. . . 130
5.5 Snapshot of the hardware setup for HIL simulation of the VSC-HVDC. 131
5.6 Block layout of the hardware setup for HIL simulation of the VSC-HVDC. 132
5.7 OP5110 FPGA module in the PCIX slot of Target node. 133
5.8 OP5130 FPGA module in the carrier slot of Target node. 134
5.9 I/O connections between the simulator and the controller. 135
5.10 PWM Generation inside FPGA 138
5.11 Oscilloscope traces for the PWM carrier, reference signal and gate pulse. 139
5.12 The s-function (a) input and output terminals (b) flow chart. 140

5.13 The truth table for event detection algorithm. 141
5.14 Experimental setup for the VSC-HVDC system. 144
5.15 Experimental Hardware: (a) reactor (b) converter used in the experiment. 145
5.16 Snapshot of the hardware setup for the VSC-HVDC experiment. . . 147
5.17 Off-line simulation results of steady-state DC-link voltage, V_{dc}, phase-a current, i_{a2}, line-to-line voltage, v_{ab2}, and phase voltage, v_{a2}. 149
5.18 Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, V_{a2}, line-to-line voltage, V_{ab2}, and phase-a current, I_{a2} and FFT of the current I_{a2} for HIL simulation. 150
5.19 Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, V_{a2}, line-to-line voltage, V_{ab2}, and phase-a current, I_{a2} and FFT of the current I_{a2} obtained from the experiment. 150
5.20 Off-line Simulation, HIL simulation and experimental results of steady-state DC-Link voltage, V_{dc}, VSC 1 power, P_1 and Q_1, and VSC 2 power, P_2 and Q_2 for the case when all reference powers are positive. 151
5.21 Fluke measurements from the experiment (a) Power, voltage and current (b) Phasor diagram. 152
5.22 Off-line simulation results of steady-state DC-Link voltage, V_{dc}, phase-a current, i_{a2}, line-to-line voltage, v_{ab2}, and phase voltage, v_{a2}. 154
5.23 Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, v_{a2}, line-to-line voltage, v_{ab2}, and phase-a current, i_{a2} and FFT of i_{a2} for HIL simulation. 155
5.24 Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, v_{a2}, line-to-line voltage, v_{ab2}, and phase-a current, i_{a2} and FFT of i_{a2} obtained from the experiment. 155
5.25 Off-line simulation, HIL simulation and Experimental result of steady-state DC-Link voltage, V_{dc}, VSC 1 power, P_1 and Q_1, and VSC 2 power, P_2 and Q_2. 156
5.26 Fluke measurements from the experiment for negative reference power (a) Power, voltage and current (b) Phasor diagram. 157

5.27 Oscilloscope trace of supply voltage in the laboratory. 158
5.28 Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, v_{b2}, line-to-line voltage, v_{ab2}, and phase-a current, i_{a2} and FFT of i_{a2} for HIL simulation using reconstructed distorted supply voltage (For positive power). 159
5.29 Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, v_{a2}, line-to-line voltage, v_{ab2}, and phase-a current, i_{a2} and FFT of i_{a2} for HIL simulation using reconstructed distorted supply voltage (For negative power). 160
5.30 Off-line simulation, HIL simulation and experimental results of DC-Link voltage, V_{dc}, VSC 1 reactive power, Q_1, and VSC 2 power, P_2 and Q_2 for changes in the DC-link voltage reference. 161
5.31 Oscilloscope trace of transient responses in DC-link voltage, V_{dc} and line-to-line voltage, v_{ab2} for HIL-simulation. 162
5.32 Oscilloscope trace of transient responses in DC-link voltage, V_{dc} and line-to-line voltage, v_{ab2} for HIL-simulation. 163
5.33 Off-line simulation, HIL simulation and experimental results of DC-Link voltage, V_{dc}, VSC 1 reactive power, Q_1, and VSC 2 power, P_2 and Q_2 when a load is added in the DC-link. 164
5.34 Off-line simulation, HIL simulation and experimental result of DC-Link voltage, V_{dc}, VSC 1 reactive power, Q_1, and VSC 2 power, P_2 and Q_2 when power references were changed. 166

A.1 (a) Resistive network (b) Inductive Network (c) Capacitive network . 186
A.2 (a) RL network (b) LC Network (c) CR network (d) RLC network . . 190

B.1 Simulation of VSC-HVDC using PSCAD/EMTDC 193
B.2 Controller 1 of VSC-HVDC in PSCAD/EMTDC 194
B.3 Controller 2 of VSC-HVDC in PSCAD/EMTDC 195
B.4 Simulation of VSC-HVDC using PSB/SIMULINK 196
B.5 Controllers of VSC-HVDC in PSB/SIMULINK 197

C.1 SIMULINK diagram for the implementation of the controllers for both HIL simulation and experimental validation of VSC-HVDC system. . 199
C.2 SIMULINK diagram of Controller 1. 200
C.3 Generation of PWM using XSG block and FPGA carrier board. . . . 201
C.4 Simulink and RTLab blocks for HIL-Simulation of VSC-HVDC. . . . 202

Chapter 1

Introduction

Rapid industrialization and subsequent increased demand of electrical energy worldwide has led to various technical problems such as system instability, degraded power quality and voltage problems in the system. In addition, limited resources, environmental constraints, deregulation and increased competition compelled utilities to come out with innovative solutions to maximize the use of their existing facilities and resources within the technical and economic limit. Both in developed and rapidly developing countries, demand continues to grow for higher level of voltages and high quality power supply. To alleviate these problems, interconnected grids with better economy, less reserve capacity and improved network reliability can be built across national frontiers by using selective new technologies. Innovations in the area of high power semiconductor devices produced powerful Thyristors, Gate Turn-Off thyristors (GTO), and Insulated Gate Bipolar Transistors (IGBT). Significant applications of these devices in power systems are High Voltage DC (HVDC), Flexible AC Transmission systems (FACTS), Variable Speed Drives (VSD) and Custom Power Systems (CPS) for transmission, distribution and utilization application. For long-distance transmission, High Voltage Direct Current (HVDC) technology has proved to be more efficient than AC transmission [1].

Conventional HVDC technology employs line commutated converters, which depend on the ac voltage for successful commutation, and commutation failure due to the in-feed of a weak AC system is not uncommon [2]. Line commutated converters may also experience problems such as slow fault recovery process and second harmonic resonance, high over-voltages following load shedding, voltage instability problems and absorption of reactive power from the source [2–5]. Despite the use of sophisticated control systems with some compensating devices, line commutation is not yet free from these inherent limitations. On the other hand, self commutated devices such as GTO or IGBT comes with the built-in turning ON/OFF capability inside the semiconductor devices and therefore, are less dependent on the availability of reliable and stiff voltage support. Self-commutated devices when used as converters are even capable of supplying the reactive power to the ac system rather than absorbing reactive power from the system [4]. Voltage Source Converters (VSC) are built using these devices and then applied in HVDC application, which is known as VSC-HVDC. The idea of using VSC for HVDC transmission is very recent and is expected to gain widespread usage [6–8].

The evolving complexity of power electronics apparatus in systems like VSC-HVDC and its dynamics with AC-DC interaction needs extensive study through simulation with maximum accuracy. The involvement of controllers for these power electronics converters made real-time simulation an important requirement to reduce cost and to evaluate the behavior of the system under certain contingency situations [9]. Real-time simulation, especially, HIL simulation is a very effective tool for comprehensive control verification. RTLAB [13–17], RTDS [10–12], HYPERSIM [18–20], dSPACE [21–23] (mostly used for real-time control), ARENE [24] and NETOMAC [25, 26] are a few leading commercial real-time simulators/simulation software used in both industry and academia for simulating various types of electrical networks.

1.1 Motivation for this Work

As power systems rely more and more on different equipment to regulate power flow, to protect the systems against any contingency situation, testing of these equipment is necessary during the design phase for their practical implementation in the system. These tests are needed to verify equipment's self behavior and its interaction with the rest of the power system. The benefits that can be achieved by using a simulator are:

- Design and testing of new apparatus and schemes
- Closed loop behavior study of the device on the full system.
- Study of the high-frequency phenomenon due to the introduction of different switching and disturbance actions.
- Training and education
- Online security control

To perform these studies analog simulators or Transient Network Analyzers (TNA) have been developed over a long time, however, analog simulation and testing of such technologies has many limitations such as insufficient flexibility, limitation of systems size, inability to deal with high X/R transmission line, high construction and maintenance costs. These drawbacks of the analog simulator did not make it an effective and well-accepted tool for extensive and complicated studies.

Advancement in the area of digital computers paved the way to the development of digital simulators that are fast and accurate enough to simulate high frequency phenomena in power system. Digital simulators can be classified into two classes: (1) off-line simulators and (2) real-time simulators. Off-line digital simulators are based on mathematical modeling and numerical analysis-based algorithms which may not reflect the accurate performance that would take place in real system. Because there are no specific requirements on the amount of time used to perform the off-line simulation, these simulation packages are not strictly tuned to produce the exact interactions happening in the system. However, real-time digital simulators especially, real-time Hardware-In-the-Loop (HIL) simulators will offer the advantage of having more accurate and reliable results than off-line digital simulators. Real-time HIL simulators are based on accurate mathematical models and solution techniques along with the option of testing or validating power systems or power electronics apparatus through digital hardware interfacing with the simulator. These simulators are capable of running the test in real-time and can analyze fast phenomena with required accuracy.

Off-line digital simulators have been commercially available for the last two decades and they have evolved through time. On the other hand, the real-time digital simulator is comparatively new and the area deserves more attention to exploit the benefit of modern, highly powerful computers. Off-line simulations can be easily performed in a standard computer as it is not restricted by any time limitations. On the other hand, real-time digital simulators must perform all the necessary calculations within the time as it happens in real life. The processing power required to solve the system equations is immense and a parallel processor based technology can be a solution to achieve real-time simulation. Thus, to overcome the limitations of off-line simulation studies and to exploit the development of the computer industry, adoption of real-time simulation is a timely step.

Real-time simulation of VSC-HVDC faces numerous challenges. The most important of which are mentioned here:

1. *Numerical Approximation Error:* The continuous nature of the power system's dynamic behavior is represented by differential equations which are eventually converted to difference equations through numerical discretizing techniques. Most of these techniques are based on the truncated Taylor's series and create numerical errors in the simulation. These errors could lead to numerical oscillation under certain conditions owing to the neglected higher order numerical terms. EMTP uses the Trapezoidal Rule to discretize the network elements which exhibits this oscillation at some situations.

2. *Time-Consuming Detailed Modeling of Power Electronics:* Simulation of power electronics in detail mode requires a longer simulation time which is not suitable for real-time simulation. Such a model include switching losses, switching transients, voltage current overshoot and detailed non-linear characteristics. An alternate fast but accurate modeling technique is required to ensure real-time simulation.

3. *Switching Delay:* The behavior of power electronic apparatus such as the HVDC link is dominated by unspecified switching discontinuities with intervals as small as a few microseconds and the inaccurate detection of such discontinuities

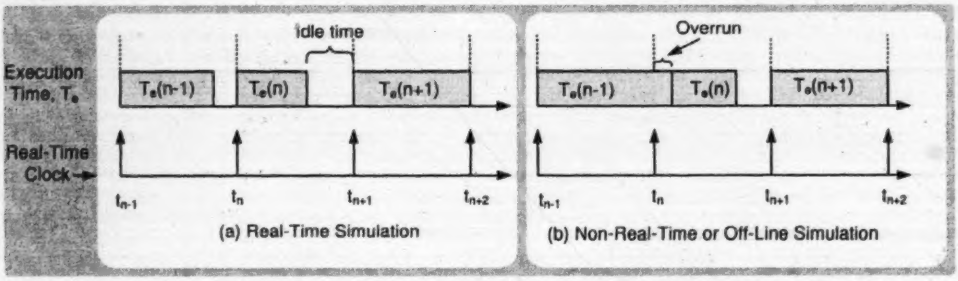
produces numerical oscillations [27]. As their occurrences do not coincide with the discrete time intervals used by the fixed time-step, the technique suffers from inaccuracies. Switchings coming in between the discrete time-points create switching delays. This problem is aggravated when multiple switchings take place within one time-step [28]. Depending on the switching frequency and simulation time-step, more than one switching events can take place in one time-step. Furthermore, since a switching event does not always come at the same time instant, the switching delay introduced is not constant; the delay can range from a fraction of a time-step up to a full time-step. Systems with power electronic switching devices need to deal with switching events that arise within the time-step.

This thesis addresses all of the aforementioned problems with an objective to produce accurate and faithful simulation for power electronics and power systems. Several fixed-step based interpolation algorithms are proposed and applied to correct the inaccuracy due to switching delays in the simulation. Improved discretization techniques are proposed and their applications under different circumstances are investigated. A general purpose processor based state-of-the-art real-time digital simulator using Commodity-Of-The-Shelf (COTS) components is interfaced with a digital controller through Field Programmable Gate Array (FPGA) based I/Os. Finally, an experimental setup was developed to validate the real-time simulation results.

1.2 Real-Time Hardware-In-the-Loop (HIL) Simulation

The term "real-time" has traditionally been used by the computer industry to refer to interactive systems where the computer response is sufficiently fast enough to satisfy human users. A more rigorous definition is applied to digital control schemes where the computer response must occur at specific time intervals. In the case of power system simulation, this implies that the computer must solve the model equations within the model time step [29]. In general, *real-time digital simulation* may be defined as a faithful reproduction of output waveforms, by combining systems of hardware and software, that would be identical to the waveforms or effects produced by the real power system being modeled.

Unlike TNAs, a digital simulator produces outputs at discrete time intervals. The system states are computed at certain discrete times either fixed or variable (used only in off-line simulation). Depending on the time taken by the computer to complete the computation of state outputs for each time-step two situations can arise. As shown in Fig 1.1(a), if the execution time, T_e, for the simulation of any time-step is smaller or equal to the time-step used, the simulation is said to be real-time simulation. On the other hand, as shown in Fig 1.1(b), if T_e for any time-step is greater than its time-step, the simulation is said to be non-real-time or off-line and in that case, execution time overruns take place. If such a situation is observed, the simulation time-step should be increased or the system model should be modified to fit the execution time within the time-step.



(a) Real-Time Simulation

(b) Non-Real-Time or Off-Line Simulation

Figure 1.1: (a) Real-Time and (b) Non-Real-Time or Off-Line Simulation.

Real-time digital simulation is a state-of-the-art technique for simulation of power systems and their components. During the last ten to fifteen years, significant efforts have been made to develop real-time digital simulators of power system networks. Developments of high speed computers and other devices accelerated the research in this area. The approach in digital simulation provides accuracy in component modeling and flexibility in component interconnection for representation of a power system. Real-time simulation simulates the power system and its components in an EMTP-type environment using time-domain solution of bus voltages and branch currents. The system is modeled with the help of a software using graphical interface on a workstation or a personal computer (PC) and then simulated on a powerful parallel processor based PC cluster. The simulation proceeds with the device-under-test connected through input and output interfaces such as filters, Digital-to-Analog (D/A) and Analog-to-Digital (A/D) converters, signal conditioners etc. The simulation can also be modified with the user defined control inputs, for example closing or opening of switches to connect or disconnect the components in the simulated power system.

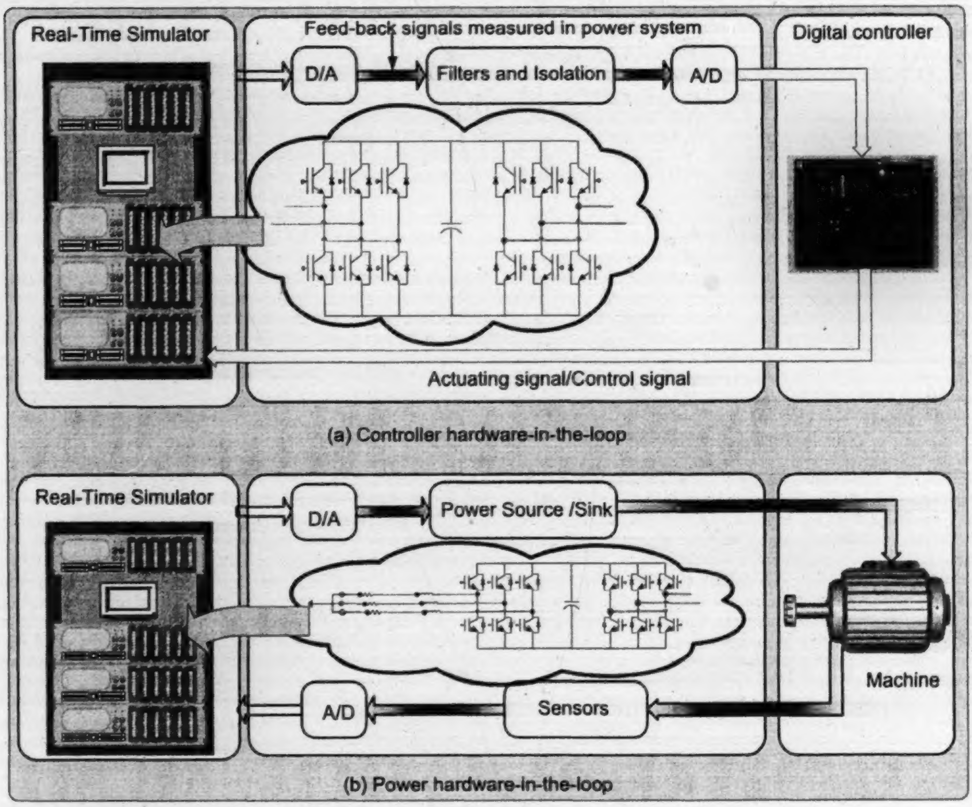

Real-Time Simulator
Feed-back signals measured in power system
Digital controller
D/A
Filters and Isolation
A/D
Actuating signal/Control signal
(a) Controller hardware-in-the-loop
Real-Time Simulator
D/A
Power Source /Sink
Machine
A/D
Sensors
(b) Power hardware-in-the-loop

Figure 1.2: HIL simulation (a) Control hardware-in-the-loop (b) Power hardware-in-the-loop.

Real-time simulation can be classified into two categories: (1) Fully digital real-time simulation and (2) Hardware-In-the-Loop (HIL) real-time simulation. A fully digital real-time simulation requires the entire system (including control, protection and other accessories) to be modeled inside the simulator and the simulation to be completed within the specified time-step. In this type of simulation, no I/Os or external interfacing is necessary (except those required for monitoring the simulation results). On the other hand, Hardware-In-the-Loop (HIL) simulation refers to a simulation, where parts of the fully digital real-time simulation have been replaced with actual physical components. Based on the replacement of the simulation by real hardware, HIL simulation can be classified into two categories. If the HIL system replaces the system under test with a real-time simulation model and control technique in the simulation model is replaced with real hardware that interacts with the simulator where the system is modeled, it is called *Controller Hardware-In-the-Loop* (CHIL). This method is also known as *rapid prototyping.* In this method, no real power transfer takes place. All of the power system and power electronic parts are modeled as a virtual system inside the simulator and external controller hardware sends controller outputs to the simulator. Therefore the full system is at a low voltage and exchanges digital and/or analog signals through the interface. Generally, a newly designed/developed controller is tested using this method where controller takes feedback signals from the simulator and processes them to produce the required output signals which are then sent back to the system (inside the simulator). Such a setup of a controller prototyping or controller HIL real-time simulator is given in Fig. 1.2(a) where a power electronic converter is modeled inside a PC-cluster and the real controller is connected to it through I/Os. If any HIL simulation involves power transfer to or from the hardware under test, it is called *Power Hardware-in-the-Loop* (PHIL). In this case, part of the power system is internally simulated and the other part is the real hardware power apparatus connected externally. In this type of simulation, a power source or sink is needed which will either generate or absorb power. An example of such a simulator could be real-time testing of machines as shown in Fig. 1.2(b), relays [30,31] or real-time testing of remotely located electrical equipments.

Fully digital simulation is often used for the understanding of behavior of a system under certain circumstances resulting from external or internal dynamic influences, however, HIL simulation is used to minimize the risk of investment through the use of a prototype once the underlying theory is established with the help of fully digital real-time simulation.

1.3 Literature Review

The focus of this literature review will be the modeling and solution techniques used for electrical network's transient simulation, the switching synchronization problem for the simulation of power electronic equipment, and off-line and real-time digital simulators developed so far.

1.3.1 Modeling and Solution Techniques

Development of a suitable and accurate simulation model is the foundation of any simulation study. To study the behavior of an electrical power system, a detailed and precise simulation model can be used which requires considerable computation time. On the other hand, real-time simulation requires fast simulation and use of simplified model with an acceptable accuracy may be the answer. A compromise must be done somewhere so that real-time simulation can be achieved without sacrificing accuracy. Reliable mathematical models and numerical techniques for time domain simulation of the electromagnetic transient analysis of power system have been available for some time [32]. Off-line simulators like EMTP and EMTDC are based on such mathematical models. Researchers have been trying to use similar or improved mathematical models and numerical techniques to develop real-time simulators. The mostly used method for the modeling and simulation of power systems are *EMTP-type* or *Numerical Integrator Substitution* method.

The electrical system is a continuous system and its variables are also continuous. But digital simulation is discrete by nature and therefore, development of suitable methods for the solution of the differential and algebraic equations at discrete points are necessary. A continuous function can be simulated by substituting a numerical integration formula into the differential equation. In the late 1960s, H.W. Dommel developed a discretization technique [33] of electrical components and variables for the efficient analysis of electromagnetic transients of power systems. The method known as EMTP-type is based on difference equations obtained through the application of the Trapezoidal Rule. The method discretizes individual components and then uses linear relationships between the currents and the voltages to obtain the nodal equations which are solved at each time-step. The solution technique is known as *Nodal Method*. The Trapezoidal integrator has been used widely for its simplicity, stability and reasonable accuracy, however, being based on a truncated Taylor's series it causes numerical oscillations under certain conditions due to the neglected terms.

State Variable Analysis is another widely used technique for the solution of differential equations. State variable formulation represents the system through a set of first-order differential equations, which are then solved by numerical integration. Forward Euler, Trapezoidal or any other integration technique can be employed to obtain the difference equations from the state equations. However, similar to Dommel's method, they are also dependant on the time-step, $\triangle t$, and any change in $\triangle t$ would need the re-computation of matrices associated with the formulation. Therefore, in cases where variable time-step simulations are involved, simulation through Trapezoidal and Forward Euler's are not suitable. Instead, some other methods such as Predictor-Corrector method or Adams Second Order method can be used. In [34], a technique for solving state-space equations using Predictor-Corrector method has been described. Instead of rearranging the state-equations into an explicit form, it uses a technique such that the state equation predict the state variable derivative and the Trapezoidal Rule corrects the estimates. In this case, calculation of state variables at times t requires information on the state variable derivatives at that time. As an initial prediction, the derivatives at the previous time-step are used.

Another technique known as Adam's Second-Order closed formula has been used in [41] to solve the state-space equations. The numerical stability of this technique is identical to that of the Trapezoidal Rule. In this method, $\triangle \mathbf{x}$ (the difference between $\mathbf{x}(t - \triangle t)$ and $\mathbf{x}(t)$) is estimated using the value of $\mathbf{x}(t - \triangle t)$ in the state equation. i.e.

$$\triangle \mathbf{x} = [\mathbf{A}\,\mathbf{x}(t - \triangle t) + \mathbf{B}\,\mathbf{u}(t)]\triangle t \tag{1.1}$$

Then the state vector is estimated as follows:

$$\mathbf{x}_{e1}(t) = \mathbf{x}(t - \triangle t) + \triangle \mathbf{x} \tag{1.2}$$

Now $\triangle \mathbf{x}$ will be re-calculated using $\mathbf{x}_{e1}(t)$ in (1.1) instead of $\mathbf{x}(t - \triangle t)$. This will yield a new $\mathbf{x}_{e2}(t)$ which will be used to re-evaluate $\triangle \mathbf{x}$ again, until there is negligible change in the updated $\mathbf{x}_{en}(t)$. It is to be noted that no matrix inversion is involved but an iterative procedure is required. Whereas using Trapezoidal Rule would involve only one evaluation instead of iteration but would require a matrix inversion when state matrices are changed due to the change of time-step.

Another part of the simulation involving power electronic devices is the precise modeling of switching devices such as the power diode, thyristor, Gate Turn-Off Thyristor (GTO) and the Insulated Gate Bipolar Transistor (IGBT). Among these, the *diode* is the only two terminal, uncontrollable device and the rest are three terminal controllable devices. To study transient simulation, it is acceptable to model a diode as a switch with its ideal characteristics. A *thyristor* is modeled by adding a turn-on control on the simplified diode model [36]. If the control is continuous, the switch simulates the diode which allows unidirectional current flow when the switch is forward biased. Delaying the gate pulse allows control over the turn-on instant of the forward biased switch. If the gating power requirement of the device is not considered, there is a very little difference between modeling a GTO, IGBT or any other three terminal, controllable, unidirectional current flowing device. All the devices can be well represented by a simplified switch with gate turn-on and turn-off controls.

Based on the modeling complexity involved, modeling of power electronic devices can be classified into two types: detailed mode and behavioral mode [38]. In the detailed-mode simulation, detailed device models are used, and the simulated circuit is constructed to resemble the actual circuit as closely as possible. Detailed mode simulation allows accurate analysis of switching characteristics and power losses. On the other hand, behavior mode simulation uses idealized or simplified device models. Only the external behavior of the device is modeled and the detailed characteristics are ignored. Since detail mode involves exact modeling and detail characteristics, it is time consuming and may not be suitable for real-time simulation. On the other hand, because of simpler model, the simulation is faster using behavioral mode and can be used in real-time simulation. Depending on the objective and need of the simulation, behavior mode simulation can be performed at three levels: *ideal device models*, *switching function model* and *averaged model*.

In *ideal-model*, the detail device characteristics are replaced by ideal or external characteristics and switching transients, diode reverse recovery, snubber circuits and stray components are omitted. In *switching function* approach, a switch converter circuit is replaced by a circuit consisting of only controlled voltage and current sources. The simulation is much faster using the switching-function models with almost the same results as the ideal-model approach gives, i.e., high-frequency effect is included. However, since individual switches no longer exist, it is not possible to monitor the voltage/current or conduction of individual switches. The *averaged model* can be obtained directly from the switching functions but for those circuits in which these functions cannot be easily obtained, the averaged model can be derived using the state-space averaging technique [39]. Since there is no switching and the circuit topology is invariant, the simulation speed can be extremely fast.

1.3.2 Switching Synchronization

Various techniques [40] are proposed in the literature to deal with *single* inter-step switching events in digital simulation under both off-line and real-time conditions. One solution for this problem was proposed in [41] is to use a variable time-step so that if switching is detected, the program changes to small time-step and reverts to the original time-step after that. Another technique performs the simulation in two time-steps when the switching events take place [42]. It uses gradual change in the resistances through the implementation of v-i characteristics of the device instead of a sudden on/off of resistances due to the switchings. Because of its involvement with the iterative process, it maybe used for off-line simulation. An alternative way is to use linear interpolation [40, 43, 44] to the point of switching, updating the model and then continuing normal simulation. However, in this technique, the time grid is no more uniformly spaced and to reach to the time grid again, several techniques are proposed in the literature [40].

In [43], a fixed step simulation technique is described which does the necessary interpolation to take into account of switching delays. Assuming instantaneous switching the states are linearly interpolated and then a method of calculating the initial conditions following every discontinuity or switching is presented. The algorithm is applied using both nodal and state-space approach. An interpolation method using instantaneous solution is described in [44]. Following a switching instant, the technique linearly interpolates all the node voltages and currents back at the switching point. The GTOs or thyristors change branch impedance (on or off) and then the network solution is repeated giving new voltages at that point. This yield two solutions (one before the switching and another after the switching) at that instant. The integration process is then performed with a full time step using that as a history value.

HYPERSIM [45], a parallel processor based real-time simulator implements a backward-forward interpolation technique to simulate switching phenomenon in power electronic simulation. In [46], a structure changing concept is discussed, which can be used to simulate the structural changes at arbitrary time instants. The technique uses both EMTP and state variable solution to accommodate any structure change due to the switching of power electronics devices. PSCAD/EMTDC uses an interpolation based algorithm to allow correction for switching delays [47]. Real-time simulator ARENE uses linear interpolation to bring the solution to the switching point, however, the interpolated values are used as the value of next time-step so that equal spacing between data points are maintained. An extrapolation is then used to bring back the solution to the original time-grid [34,47]. A method based on re-initialization through interpolation and extrapolation for treatment of discontinuities in simulating power electronic circuits has been proposed in [48].

1.3.3 Off-Line Electromagnetic Transient Simulation

The EMTP has become an industrial standard and was enriched from contribution from many people. In 1982, EMTP Development Co-ordination Group (DCG) was formed to document, maintain and support the program, in an agrement with Electric Power Research Institute (EPRI). Their joint effort produced the EPRI-DCG EMTP version 1.0 in 1987 and version 2.0 in 1989. Since 2000, Hydro-Quebec acquired the rights to rewrite and commercialize the EPRI-DCG EMTP under a new name called EMTP-RV. Another version of EMTP known as the Alternative Transient Program (ATP) produced by a group of professionals in order to make EMTP accessible to the worldwide community. Major characteristics of that version was the inclusion of Transient Analysis of Control System (TACS), frequency dependant transmission line model, three phase transformer model, synchronous machine model and underground cable model etc.

Emerging use of power electronics converters (especially HVDC converter) in power systems motivated a group of people like D. A. Woodford of Manitoba Hydro and others to develop a program based on the EMTP concept but is capable of simulating AC-DC systems. This program known as EMTDC (Electromagnetic Transient Program for DC) originally used a mainframe computer to perform simulations. Then Manitoba HVDC research center developed a comprehensive graphical user interface called PSCAD (Power System Computer Aided Design) to simplify and speed up the simulation task in EMTDC and were released in 1990 for Unix workstations. However, following the emergence of Windows as a dominant operating system for PCs, Manitoba HVDC research center rewrote windows based PSCAD/EMTDC and released in 1998. The latest release of this program (version 4.2.1) took place in the year 2005.

The other EMTP-type programs have also faced the same situation and developed programs for windows with graphical user interface, such as ATP_Draw from ATP. More recent trend is to increase the functionality to allow integration with other programs. For example, MATLAB/SIMULINK has very rich specialized toolboxes, and a couple of the programs allow the interfacing with MATLAB to benefit from the use of such toolboxes in the transient simulation program. Following the footsteps of EMTP, many other programs are now developed and commercially available in the market. Currently, several other off-line digital simulation software tools are also available, with varying degrees of modeling and simulation capabilities, such as PSS/E, SPICE, SABER and many more.

1.3.4 Evolution of Real-Time Digital Simulator

Despite the dramatic progress in the area of increasing computation power of the PCs, most of the PC-based digital simulators were off-line. Very recently, using parallel processor-based technologies and improved numerical analysis techniques with less computational burden, quite a number of real-time simulators have been proposed and tested [10]- [26]. Initially, real-time simulators were based on DSP [11, 52–57], RISC [58–60], CISC and VLSI technology [15, 61, 62], however, the use of general-purpose processor as the computation engine has become an attractive option because of their lower cost and faster development cycle. Development of clustered systems using off-the-shelf processors and advanced communication networks is a growing trend for the development of real-time simulator now a days.

The first commercial real-time digital simulator was demonstrated by RTDS Technologies in 1991 using DSPs. It was interfaced to the controller of a HVDC converter to asses its performance. A combination of both analog and digital parts were used in that simulator. A fully digital real-time Digital Transient Network Analyzer (DTNA) that can perform real-time tests of power system equipments on standard multi-purpose parallel computers have been discussed in [60]. The DTNA can simulate phenomenon like electromagnetic transients upto 3 kHz, AC/DC interactions, electro-mechanical transients and similar long time phenomena. A wide variety of components, equipments and controllers including power electronic based controllers can be modeled and simulated using the simulator.

No fully digital real-time simulator was possible in an standard computer before 1996, when Eletric De France (EDF) introduced their first real-time simulator ARENE [24]. It was capable of simulating high-frequency phenomenon in a standard but multi-purpose parallel computer [63]. Another PC-based real-time simulator NETOMAC [25, 26] from SIEMENS has been used to simulate large power grids. Since 1996, it has been commercially used as real-time simulator for interactive testing of protection devices [64]. HYPERSIM [18–20] is another fully digital real-time simulator for the analysis of multi-phase electromagnetic and electromechanical transients in power systems. It was initially designed by Hydro-Qubec's research institute IREQ to solve complex transmission and distribution problems.

Opal-RT Technologies Inc. offers a general-purpose processor based real-time simulator which is flexible, scalable and it uses COTS components [13]. In conjunction with its real-time software RTLAB, and other modeling software such as RT-EVENTS, ARTEMIS, it uses MATLAB/SIMULINK as the main backbone for the simulation. The simulator allows the flexibility of using any custom model or solver in addition to the above-mentioned modeling programs. Almost similar approach of using standard PC-based cluster is adopted by dSPACE which uses AMD processors and MATLAB/SIMULINK as the modeling package, though their older generation of simulators used DSPs [21–23].

Other than the aforementioned real-time simulators, custom real-time simulators are also mentioned in the literature which are mostly developed and used in the laboratories. These real-time simulators use a combination of hardware and software mostly to serve specific requirements needed at times. FPGAs are another kind of digital processors that are firming their foundation in real-time simulators. User-friendly block coding, interfaced with SIMULINK allows some FPGAs to be used as another platform to model and then to simulate small systems within a time-step of nanoseconds [37]. This is another promising avenue for future exploitation in simulation industry.

1.4 Thesis Objectives

The main objectives of this thesis are:

1. To develop an accurate mathematical model of the VSC-HVDC system for electromagnetic transient simulation.
2. To develop novel computational techniques for discretization of the developed model, and to develop accurate and efficient techniques to synchronize the switching discontinuities inline with the real-time simulation.
3. To design a digital controller for the VSC-HVDC system.

4. To implement the VSC HVDC system and its controller in commercial off-line simulation software such as MATLAB/SIMULINK and PSCAD/EMTDC and to compare the simulation results.

5. To implement the developed models and techniques in a PC-cluster based real-time simulator using digital hardware and software and to interface the digital controller in the HIL setup.

6. To experimentally validate the off-line and real-time HIL simulation and control results by implementing the controller for a 4kW VSC-HVDC system.

1.5 Thesis Outline

The thesis is organized as follows:

- Chapter 2 is **Network Discretization and Solution Techniques.** This chapter introduces new applications of discretization and solution techniques for various components/circuits used in power systems. It also does a comparative study of the proposed discretization techniques with the existing ones.

- Chapter 3 is **Switching Events Synchronization**. This chapter deals with the challenges faced by the simulator when switchings occur in between two time-steps. It discusses the existing techniques for the accounting of switchings and proposes a few algorithms to tackle multiple switchings in one time-step. Proposed algorithms are applied in the simulation of a 6-pulse STATCOM and a comparative analysis of their performance is presented in this chapter.

- Chapter 4 is **Modeling, Control and Offline Simulation of VSC-HVDC**. In this chapter, issues related to modeling of the test system in PSCAD/EMTDC and PSB/SIMULINK, design and implementation of its digital controller and the off-line simulation results obtained by both programs are discussed.

- Chapter 5 is **Real-Time HIL Simulation of VSC-HVDC and Experimental Validation.** This chapter discusses the real-time modeling, control and simulation of the system. The network is first modeled for an off-line simulation using C language and later the C-program is modified to suit the real-time HIL simulation. Details about the setup for HIL simulation is then discussed. This chapter also concentrates on the experimental setup of a 4KW VSC-HVDC system. The results obtained through off-line C-program and real-time HIL simulation are then compared against the experimental results to validate the accuracy of the both off-line and real-time simulation.

- Chapter 6 presents the **Conclusions** of the thesis. Recommendations are also made for future research work.

Chapter 2

Network Discretization and Solution Techniques

2.1 Introduction

Development of a discrete-time model is a requirement for the digital simulation of electromagnetic transients in electrical systems. Linear dynamics of the power systems are generally expressed through continuous-time equations that needs to be approximated through discrete-time models. Even though various discretization techniques are available [33–35], there is no unique, accurate and stable equivalence between the continuous-time and discrete-time systems. The most commonly used discretization techniques for electromagnetic transient simulation are based on numerical integration algorithms. However, another technique which has been used in the design of digital controllers and digital filters [67], can also be applied to perform the digital simulation of power systems. This technique uses Z-domain transformation to produce the difference equations needed for time-domain simulation of systems. The linear difference equations which are algebraic in nature can be solved for any given input(s) and the response(s) can be determined. This chapter[1] attempts to develop accurate discretization techniques and thus concentrates on:

- Deriving discretized Norton Equivalents for various electrical components used in the circuits, using Step-Invariant and Ramp-Invariant Transformation.

---

[1] Material from this chapter is under preparation for publication: M. O. Faruque, V. Dinavahi, "Step-Invariant and Ramp-invariant Transformations for Transient Simulation of Electrical Networks", to be submitted to the *IEEE Trans. on Power Delivery*.

- Applying these equivalents for electro-magnetic transient simulation.
- Comparing these techniques with the established discretizing methods and highlighting their improved accuracy over existing methods.

2.2 Step-Invariant and Ramp-Invariant Transformation

Z-transform is a very powerful operational method in discrete-time analysis which considers only the sampled values of a function $x(t)$, i.e., $x(0)$, $x(\triangle t)$, $x(2\triangle t)$, ..., where $\triangle t$ is the sampling period. The Z-transform of a function $x(t)$ (where t is non-negative) or of a sequence of values of $x(k\triangle t)$ (where k is zero or positive integers) is defined by the following equation:

$$\begin{aligned} X(z) &= \mathcal{Z}[x(t)] = \mathcal{Z}[x(k\triangle t)] \\ &= \sum_{k=0}^{\infty} x(k\triangle t) z^{-k} \\ &= x(0) + x(\triangle t)z^{-1} + x(2\triangle t)z^{-2} + \cdots\cdot + x(k\triangle t)z^{-k} + \cdots\cdot \end{aligned} \tag{2.1}$$

Equation (2.1) implies that the Z-transform of any continuous time function can be written in the form of a series, where z^{-k} indicates the position in time at which the amplitude is $x(k\triangle t)$. The inverse of Z-transform of these sequence of $x(k\triangle t)$ will give the values of $x(t)$ at the respective instants of time. Thus, it gives only a time sequence that specifies the values of $x(t)$ at discrete instants of time, $t = 0, \triangle t, 2\triangle t, \cdots$ and says nothing about the values of $x(t)$ at all other times. If $x(t)$ is the response of a system with transfer function $G(s)$ and if its input is $u(t)$, digital simulation will yield responses, $x(k\triangle t)$ only at time, $t = 0, \triangle t, 2\triangle t, \cdots\cdots$. However, based on the assumptions of variations in inputs between any two consecutive discrete points, the response may not be the same at a particular time instant. In the area of discrete control, systems are discretized for design purpose where input $u(t)$ is assumed to vary either as a step function or a ramp function between two discrete instants (as shown in Fig. 2.1). These discretizations are known as *Step-Invariant Transformation* (SIT) and *Ramp-Invariant Transformation* (RIT), respectively [67].

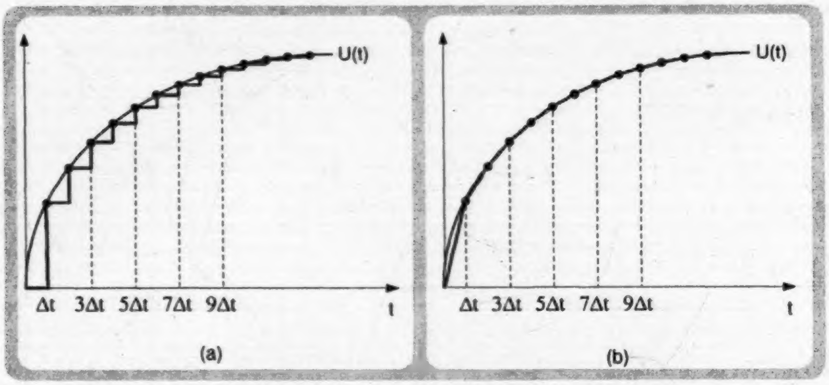


Figure 2.1: (a) Step-Invariant input (b) Ramp-Invariant input.

2.2.1 Transfer Function Approach

If it is considered that during the sampling interval, the input is held constant at the value it had at the beginning of the interval, the input is considered as step-invariant. Therefore, instead of a discrete value, the input to the system would be a step function and the overall transfer function is the cascade combination of a ZOH (Zero-Order Hold) and the transfer function of the plant. Physical systems which are linear and time invariant, can be represented as shown in Fig. 2.2(a) with an input $u(t)$, a transfer function $G(s)$ and an output $y(t)$. However, to perform computer simulation of this system, its digital equivalent shown in Fig. 2.2(b) is necessary where the input is $u(k\triangle t)$, the transfer function is $G(z)$ and the output is $y(k\triangle t)$. The conversion can be done using step-invariant transformation (Fig. 2.2(c)), which is equivalent to adding a ZOH at the beginning of the plant and an A/D converter after the plant. The mathematical basis of this conversion is given below. The Laplace transform of a pulse $p(t)$ with height 1 and width $\triangle t$ is computed as $\frac{1}{s}(1-e^{-s\triangle t})$. If the input of a ZOH is 1, then the output will be $p(t)$. Therefore, the ZOH can be considered to have a transfer function $G_h(s) = \frac{1-e^{-s\triangle t}}{s}$. The cascaded transfer function of the ZOH and the plant is $G(s)\frac{1-e^{-s\triangle t}}{s}$. The digital input sequence of $u(k\triangle t)$ can be represented by

$$u^*(t) = \sum_{k=0}^{\infty} u(k\triangle t)\delta(t - k\triangle t) \qquad (2.2)$$

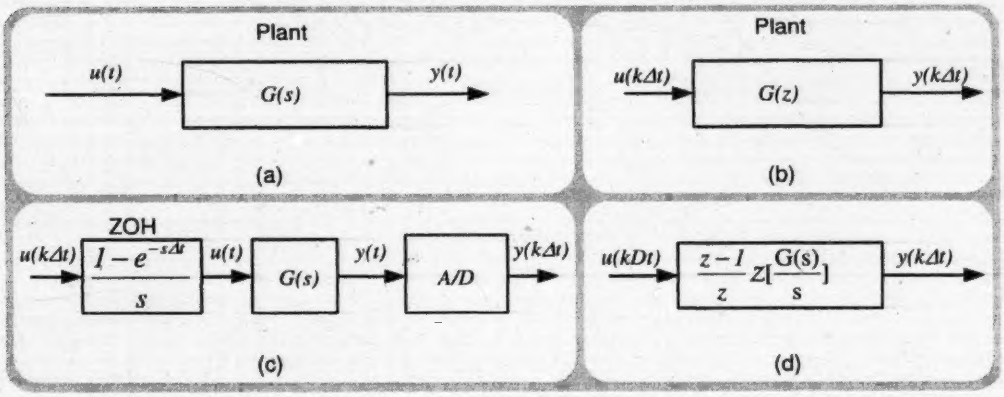


Figure 2.2: (a) Analog plant (b) Required digital plant for simulation (c) Conversion to digital plant (d) Converted digital plant

Now considering $y(k\Delta t)$ as the output for an input $u(k\Delta t)$, the ZOH and the analog plant can be considered as an equivalent digital system with discrete transfer function $G(z)$;

$$G(z) = \mathcal{Z}\left[(1 - e^{-s\Delta t})\frac{G(s)}{s}\right] = \mathcal{Z}\left[\frac{G(s)}{s}\right] - \mathcal{Z}\left[e^{-s\Delta t}\frac{G(s)}{s}\right] \tag{2.3}$$

As $e^{-s\Delta t}$ introduces a time-delay only, it can be taken outside the z-transform

$$G(z) = \mathcal{Z}\left[\frac{G(s)}{s}\right] - e^{-s\Delta t}\mathcal{Z}\left[\frac{G(s)}{s}\right] = (1 - e^{-s\Delta t})\mathcal{Z}\left[\frac{G(s)}{s}\right] \tag{2.4}$$

Using $z = e^{s\Delta t}$, the above equation can be found as,

$$G(z) = (1 - z^{-1})\mathcal{Z}\left[\frac{G(s)}{s}\right] \tag{2.5}$$

which can be written as,

$$G(z) = \frac{z-1}{z}\mathcal{Z}\left[\frac{G(s)}{s}\right] \tag{2.6}$$

This approximation of $G(z)$ is called *Step-Invariant Transformation (SIT)*, which yields exact response only when the input is actually a step function, or a sequence of steps, with changes occurring only at the sample instants. For other inputs, the approximation will introduce some errors in the solution. Given a sampling rate, all transformations involved in (2.6) are well defined and the conversion will exist and is unique.

Thus, G(z) can be determined in a unique and simple way for a given G(s) and the response $y(t)$ can be obtained from the following equation:

$$y(t)|_{t=k\triangle t} = \mathcal{Z}^{-1}[G(z)U(z)] \tag{2.7}$$

If the s-domain transfer function is strictly proper, step-invariant transformation will introduce at least one pole in the origin which creates a pure time-delay in the time-domain representation of the outputs. This produces an *explicit* algorithm which could be useful for the implementation of real-time simulation and control. However, the delay can be compensated through adding zero(s) at origin in the Z-domain transfer function, G(z). In that case, the response $y(t)$ may have minimum phase delay and the transformation may yield an *implicit* algorithm which can be given as:

$$y(t)|_{t=k\triangle t} = \mathcal{Z}^{-1}[z^r G(z)U(z)] \tag{2.8}$$

where r is the number of zeros added. In this thesis, we call this transformation *Time Shifted Step-Invariant Transformation (TSSIT)*. Generally, an *implicit* algorithm is not suitable for real-time simulation as the output, $y(k\triangle t)$ at time $t = k\triangle t$ depends on $u(k\triangle t)$. However, in most cases $u(k\triangle t)$ is predictable and TSSIT can be easily evaluated. This will need an error checking stage in the simulation and corrections would be necessary if there is an error between the predicted input and the actual input.

Similarly, if the input $u(t)$ is a ramp function during a sample interval, $n\triangle t \leq t < (n+1)\triangle t$, then the transformation is known as *Ramp-Invariant Transformation (RIT)* . Using a First-Order Hold (FOH) instead of a ZOH in Fig. 2.2, a ramp invariant transformation can be realized. The final form of the equation for evaluating the ramp invariant transformation from the transfer function is given as follows:

$$G(z) = \left[\frac{(1-z^{-1})^2}{\triangle t z^{-1}}\right] \mathcal{Z}\left[\frac{G(s)}{s^2}\right] \tag{2.9}$$

For both of these approximations (SIT and RIT), the poles of $G(z)$ and $G(s)$ are related through the simple transformation $z = e^{s\triangle t}$ but the residues are different [65]. Z-transform of $G(s)/s$ or $G(s)/s^2$ is obtained through expressing them in the partial fraction form with their poles in s-domain and then substituting by their discrete equivalent in terms of z. For example, consider a function $f(t) = e^{-at}$. The Laplace transform of $f(t)$ is $F(s)$ and given by

$$F(s) = \mathcal{L}[f(t)] = \mathcal{L}[e^{-at}] = \frac{1}{s+a} \tag{2.10}$$

Now, if the z-transform of $f(t)$ is $F(z)$, in the discrete domain $f(k\Delta t) = e^{-ak\Delta t}$ and

$$F(z) = \mathcal{Z}[f(k\Delta t)] = \mathcal{Z}[e^{-ak\Delta t}] \tag{2.11}$$

Using the basic definition of z-transform from (2.1),

$$F(z) = \sum_{k=0}^{\infty} e^{(-ak\Delta t)} z^{-k}$$

$$\tag{2.12}$$

$$= \sum_{k=0}^{\infty} \left[\frac{e^{-a\Delta t}}{z}\right]^{k} \tag{2.13}$$

$$\tag{2.14}$$

$$= 1 + \left[\frac{e^{-a\Delta t}}{z}\right] + \left[\frac{e^{-a\Delta t}}{z}\right]^{2} + \left[\frac{e^{-a\Delta t}}{z}\right]^{3} \cdots + \left[\frac{e^{-a\Delta t}}{z}\right]^{n} + \cdot\cdot \tag{2.15}$$

Using Mclaurin's series, we then find

$$F(z) = \left[1 - (\frac{e^{-a\Delta t}}{z})\right]^{-1} = \frac{1}{1 - (\frac{e^{-a\Delta t}}{z})} = \frac{z}{z - e^{-a\Delta t}} \tag{2.16}$$

Now from (2.10) and (2.16), it can be written that

$$\frac{z}{z - e^{-a\Delta t}} = \mathcal{Z}\left[\frac{1}{s+a}\right] \tag{2.17}$$

This, however, maps s-domain and z-domain by the relationship $z = e^{s\Delta t}$ and details about the mapping are available in [65, 67]. In general, transfer functions can be expressed as

$$G(s) = \sum_{i=1}^{n} \frac{A_i}{s + p_i} \tag{2.18}$$

Then the poles of $G(z)$ are obtained as $z_i = e^{p_i\Delta t}$ and for step-invariant transformation, the discrete-time transfer function is given by [65],

$$G(z) = \sum_{i=1}^{n} \frac{A_i(1 - e^{-p_i\Delta t})}{p_i(z - e^{-p_i\Delta t})} = \sum_{i=1}^{n} \frac{B_i}{z - e^{-p_i\Delta t}} \tag{2.19}$$

where, $B_i = \frac{A_i}{p_i}(1 - e^{-p_i \Delta t}), p_i \neq 0$.

For the ramp-invariant transformation,

$$G(z) = \frac{1}{\Delta t} \sum_{i=1}^{n} \frac{A_i}{p_i^2} \frac{(\Delta t p_i - 1 + e^{-p_i \Delta t})z + (1 - e^{-p_i \Delta t} - \Delta t p_i e^{-p_i \Delta t})}{z - e^{-p_i \Delta t}} = \sum_{i=1}^{n} \frac{C_i z + D_i}{z - e^{-p_i \Delta t}} \tag{2.20}$$

where,

$$C_i = \frac{A_i}{\Delta t p_i^2}(\Delta t p_i - 1 + e^{-p_i \Delta t}) \tag{2.21}$$

$$D_i = \frac{A_i}{\Delta t p_i^2}(1 - e^{-p_i \Delta t} - \Delta t p_i e^{-p_i \Delta t}) \tag{2.22}$$

For both of these transformations, a considerable amount of computation time is required, which is a challenge for the real-time simulation. Irrespective of the transformation technique used, the final form of a difference equation can be generalized as;

$$\begin{aligned} y(k\Delta t) &= a_1 y[(k-1)\Delta t] + a_2 y[(k-2)\Delta t] + a_3 y[(k-3)\Delta t] + \cdots \\ &\quad + b_0 u(k\Delta t) + b_1 u[(k-1)\Delta t] + \cdots \\ &= \sum_{i=1}^{m} a_i y[(k-i)\Delta t] + \sum_{j=0}^{n} b_j u[(k-j)\Delta t] \end{aligned} \tag{2.23}$$

where, the values of the coefficients $a_1, a_2, a_3, \cdots m$ and $b_0, b_1, \cdots n$ depends on the transfer function and the time-step, Δt, used for descretization, and the values of m and n depends on the transformation technique used.

2.2.2 State-Space Approach

The step-invariant $G(z)$ can also be obtained from analog plant $G(s)$ by using state-variable equations [66]. Let us consider

$$\dot{\mathbf{x}}(t) = \mathbf{A_c x}(t) + \mathbf{B_c u}(t) \tag{2.24}$$

$$\mathbf{y}(t) = \mathbf{C_c x}(t) + \mathbf{D_c u}(t) \tag{2.25}$$

as the realization of $G(s)$. The exact solution to (2.24) is

$$\mathbf{x}(t) = e^{\mathbf{A_c} t} \int_{t_0}^{t} e^{-\mathbf{A_c} \tau} \mathbf{B_c u}(\tau) d\tau + e^{\mathbf{A_c}(t-t_0)} \mathbf{x}(t_0) \tag{2.26}$$

where, $\mathbf{x}(t_0)$ is the value of $\mathbf{x}(t)$ at $t = t_0$. The above equation is not suitable for digital simulation. In digital simulation, $\mathbf{x}(t)$ can only be calculated at discrete values of t usually, for $t = k\Delta t$, where k is an integer and Δt is time-step. Since the input, $\mathbf{u}(k\Delta t)$ is assumed to be known for all k, let us substitute $t_0 = k\Delta t$ and $t = (k+1)\Delta t$ in (2.26). Then,

$$\mathbf{x}[(k+1)\Delta t] = e^{\mathbf{A_c(k+1)\Delta t}} \int_{k\Delta t}^{(k+1)\Delta t} e^{-\mathbf{A_c}\tau}\mathbf{B_c}\mathbf{u}(\tau)d\tau + e^{\mathbf{A_c}\Delta t}x(k\Delta t) \tag{2.27}$$

If the input, $\mathbf{u}(t)$ is step-invariant,

$$\mathbf{u}(t) = \mathbf{u}(k\Delta t), \qquad k\Delta t \leq t < (k+1)\Delta t, k = 0, 1, 2, 3, \cdots \tag{2.28}$$

then the output $y(t)$ at $t = k\Delta t$ can be described as;

$$\begin{aligned} \mathbf{x}(k+1)\Delta t &= \mathbf{A_d}\mathbf{x}(k\Delta t) + \mathbf{B_d}\mathbf{u}(k\Delta t) & (2.29) \\ \mathbf{y}(k\Delta t) &= \mathbf{C_d}\mathbf{x}(k\Delta t) + \mathbf{D_d}\mathbf{u}(k\Delta t) & (2.30) \end{aligned}$$

where, $\mathbf{A}_d = e^{\mathbf{A_c}\Delta t}$, $\mathbf{B}_d = \left(\int_0^{\Delta t} e^{\mathbf{A_c}\tau}d\tau\right)\mathbf{B_c} = [e^{\mathbf{A_c}\Delta t} - 1]\mathbf{A_c}^{-1}\mathbf{B_c}$, $\mathbf{C_c} = \mathbf{C_d}$ and $\mathbf{D_c} = \mathbf{D_d}$. The presence of $\mathbf{A_c}^{-1}$ can be avoided by using the following equation;

$$\begin{aligned} [e^{\mathbf{A_c}\Delta t} - 1]\mathbf{A_c}^{-1} &= [(1 + \mathbf{A_c}\Delta t + \frac{1}{2!}\mathbf{A}_\mathbf{c}^2\Delta t^2 + \frac{1}{3!}\mathbf{A}_\mathbf{c}^3\Delta t^3 + \cdots) - 1]\mathbf{A_c}^{-1} \\ &= 1.\Delta t + \frac{1}{2!}\mathbf{A_c}\Delta t^2 + \frac{1}{3!}\mathbf{A}_\mathbf{c}^2\Delta t^3 + \cdots & (2.31) \end{aligned}$$

However, for ramp-invariant input,

$$\mathbf{u}(t) = \mathbf{u}(k\Delta t) + \frac{\mathbf{u}[(k+1)\Delta t] - \mathbf{u}(k\Delta t)}{\Delta t} \times (t - k\Delta t) \tag{2.32}$$

where, $k\Delta t \leq t \leq (k+1)\Delta t, k = 0, 1, 2, 3, \cdots$, and the output $\mathbf{y}(t)$ at $t = k\Delta t$ can be described as [66]:

$$\begin{aligned} \mathbf{x}(k+1)\Delta t &= \mathbf{F_d}\mathbf{x}(k\Delta t) + \mathbf{G_d}\mathbf{u}(k\Delta t) + \mathbf{H_d}\mathbf{u}(k+1)\Delta t & (2.33) \\ \mathbf{y}(k\Delta t) &= \mathbf{C_d}\mathbf{x}(k\Delta t) + \mathbf{D_d}\mathbf{u}(k\Delta t) & (2.34) \end{aligned}$$

where,

$$\begin{aligned}
\mathbf{F_d} &= e^{\mathbf{A_c}\Delta t} = \sum_{n=0}^{\infty} \frac{1}{n!}(\mathbf{A_c}\Delta t)^n && (2.35)\\
\mathbf{G_d} &= [e^{\mathbf{A_c}\Delta t}(-1 + \mathbf{A_c}\Delta t) + 1](\mathbf{A}_c^2\Delta t^2)^{-1}\mathbf{B}_c\Delta t \\
&= \sum_{n=0}^{\infty} \frac{1}{n(n+2)!}(\mathbf{A_c}\Delta t)^n \times \mathbf{B_c}\Delta t && (2.36)\\
\mathbf{H_d} &= [e^{\mathbf{A_c}\Delta t} - 1 - \mathbf{A_c}\Delta t](\mathbf{A_c}^2\Delta t^2)^{-1}\mathbf{B_c}\Delta t \\
&= \sum_{n=0}^{\infty} \frac{1}{(n+2)!}(\mathbf{A_c}\Delta t)^n \times \mathbf{B_c}\Delta t && (2.37)\\
\mathbf{C_d} &= \mathbf{C_c} && (2.38)\\
\mathbf{D_d} &= \mathbf{D_c} && (2.39)
\end{aligned}$$

Both step-invariant and ramp-invariant transformation techniques are derived based on the assumption that the input $u(t)$ is either piecewise-constant or piecewise-linear as shown in Fig. 2.3. Other transformation techniques to obtain the approximate difference equations by assuming $\dot{\mathbf{x}}(t)$ in (2.24) or the integrand in (2.26) to be piecewise constant, piecewise-linear, or piecewise parabolic are also available [66].

1. If $\dot{\mathbf{x}}(t)$ in (2.24) is assumed to be piecewise-constant; i.e. $f(t) = \dot{\mathbf{x}}(t)$ in Fig. 2.3(a), discrete approximation of 2.24 can be given as:

$$\begin{aligned}
\frac{\mathbf{x}(k+1)\Delta t - \mathbf{x}(k\Delta t)}{\Delta t} &= \mathbf{A_c}\mathbf{x}(k\Delta t) + \mathbf{B_c}\mathbf{u}(k\Delta t)\\
\mathbf{x}(k+1)\Delta t &= (\mathbf{I} + \mathbf{A_c}\Delta t)\mathbf{x}(k\Delta t) + \mathbf{B_c}\Delta t\mathbf{u}(k\Delta t) && (2.40)
\end{aligned}$$

   This is also known as the Forward Euler formula. Another version of discrete approximation which is also based on piecewise-constant approximation is known as the Backward Euler approximation and is given as follows:

$$\begin{aligned}
\frac{\mathbf{x}(k+1)\Delta t - \mathbf{x}(k\Delta t)}{\Delta t} &= \mathbf{A_c}\mathbf{x}(k+1)\Delta t + \mathbf{B_c}\mathbf{u}(k+1)\Delta t && (2.41)\\
\mathbf{x}(k+1)\Delta t &= (\mathbf{I} - \mathbf{A_c}\Delta t)^{-1}\mathbf{x}(k\Delta t) + (\mathbf{I} - \mathbf{A_c}\Delta t)^{-1}\mathbf{B_c}\Delta t\mathbf{u}(k+1)\Delta t
\end{aligned}$$

2. If $\dot{\mathbf{x}}(t)$ is assumed to be piecewise-linear in (2.24) i.e., $f(t) = \dot{\mathbf{x}}(t)$ in Fig. 2.3(b),

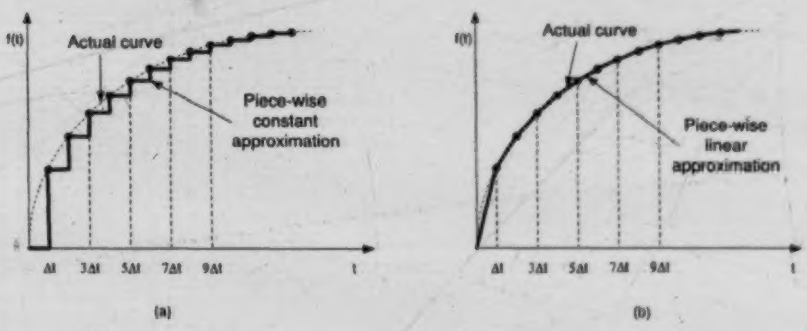


Figure 2.3: (a) Piecewise-constant integrand (b) Piecewise-linear integrand.

the discrete approximation of (2.24) can be given as:

$$\begin{aligned}\frac{\mathbf{x}(k+1)\Delta t-\mathbf{x}(k\Delta t)}{\Delta t} &= \mathbf{A_c}\left[\frac{\mathbf{x}(k+1)\Delta t+\mathbf{x}(k\Delta t)}{2}\right]+\mathbf{B_c}\left[\frac{\mathbf{u}(k+1)\Delta t+\mathbf{u}(k\Delta t)}{2}\right]\\ \mathbf{x}(k+1)\Delta t &= \left(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A_c}\right)^{-1}\left(\mathbf{I}+\frac{\Delta t}{2}\mathbf{A_c}\right)\mathbf{x}(k\Delta t)\\ &+ \left(\mathbf{I}-\frac{\Delta t}{2}\mathbf{A_c}\right)^{-1}\frac{\Delta t}{2}\mathbf{B_c}[\mathbf{u}(k\Delta t)+\mathbf{u}(k+1)\Delta t] \qquad (2.42)\end{aligned}$$

which is known as the Trapezoidal Rule.

3. If the integrand in (2.27) is considered constant, i.e. $f(t) = e^{\mathbf{A_c}\Delta t}\mathbf{B_c}\mathbf{u}(t)$ is assumed to be piecewise constant as shown in Fig. 2.3(a), we can get,

$$\mathbf{x}(k+1)\Delta t = e^{\mathbf{A_c}}\Delta t\mathbf{x}(k\Delta t)+\Delta t e^{\mathbf{A_c}\Delta t}\mathbf{B_c}u(k\Delta t) \qquad (2.43)$$

or

$$\mathbf{x}(k+1)\Delta t = e^{\mathbf{A_c}\Delta t}\mathbf{x}(k\Delta t)+\Delta t e^{\mathbf{A_c}\Delta t}\mathbf{B_c}u(k+1)\Delta t \qquad (2.44)$$

4. If the integrand in (2.27) is considered linear, i.e., $f(t) = e^{\mathbf{A_c}\Delta t}\mathbf{B_c}\mathbf{u}(t)$ is assumed to be piecewise linear as shown in Fig. 2.3(b), we can get,

$$\mathbf{x}(k+1)\Delta t = e^{\mathbf{A_c}\Delta t}\mathbf{x}(k\Delta t)+e^{\mathbf{A_c}\Delta t}\mathbf{B_c}\frac{\Delta t}{2}[\mathbf{u}(k\Delta t)]+\mathbf{B_c}\frac{\Delta t}{2}[\mathbf{u}(k+1)\Delta t] \qquad (2.45)$$

5. If the integrand in (2.27) is considered a second degree polynomial in t,

evaluation of the integrand will lead to Simpson's rules.

$$\begin{aligned}\mathbf{x}[(2n+2)\Delta t] &= e^{2\mathbf{A}_c\Delta t}\mathbf{x}(2n\Delta t) + e^{2\mathbf{A}_c\Delta t}\frac{\Delta t}{3}\mathbf{B}_c\mathbf{u}(2n\Delta t) \\ &+ e^{\mathbf{A}_c\Delta t}\frac{4}{3}\Delta t\mathbf{B}_c\mathbf{u}[(2n+1)\Delta t] + \frac{\Delta t}{3}\mathbf{B}_c\mathbf{u}[(2n+2)\Delta t]\end{aligned} \tag{2.46}$$

2.3 Accuracy of SIT and RIT

One of the main drawbacks of discrete simulation is the introduction of errors due to the numerical approximations used to discretize the differential equations obtained from the system dynamics. An understanding of errors will lead to the selection of better discrete transformation. The error analysis for both the transfer function approach and the state-space approach are discussed next.

2.3.1 Errors in Transfer Function Approach

For transfer function based step-invariant or ramp-invariant discretization technique, it is difficult or almost impossible to derive a generalized expression for the analysis of errors as it can be done for techniques or algorithms based on truncated Taylor's series. However, a standard practice of using a linear, first order test equation can be exercised to demonstrate the difference between the *local error* and the *total error*. Let us consider the equation,

$$\dot{\mathbf{y}} = f(\mathbf{y}, t) = -\lambda\mathbf{y} \tag{2.47}$$

whose exact analytical solution is $y(t) = y_0 e^{-\lambda t}$, $t \geq 0$ where $y_0 \triangleq y(0)$ is the initial condition. The reason for choosing the above equation as a test equation is not only that it has an exact analytical solution, but also most solutions to a differential equation can be approximated by a portion of an exponential. If an algorithm fails to solve this equation, chances are that it will fail when applied to other numerical equations. Since the solution through numerical simulation is never exact, errors are incurred at each time-step as well as after some prescribed time interval. In Fig. 2.4, measures of this errors are explained for the equation $\dot{\mathbf{y}} = -\lambda\mathbf{y}$ as *local error* (ε_L) at $t = t_{n+1}$,

$$\varepsilon_L \triangleq [y_n e^{\lambda\Delta t}] - y_{n+1} \tag{2.48}$$

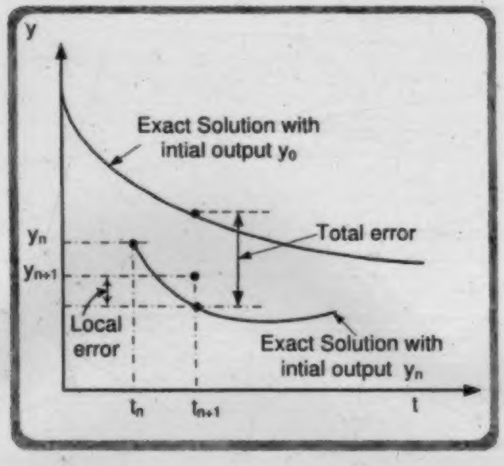


Figure 2.4: Difference between local and total errors.

and the *total error* ($\varepsilon_{\triangle}t$) at $t = t_{n+1}$,

$$\varepsilon_{\triangle}t \triangleq [y_0 e^{\lambda t_{n+1}}] - y_{n+1} \tag{2.49}$$

The *local error* (ε_L) defined by (2.48) can be defined as the error occurring at $t = t_{n+1}$, assuming that y_n is the initial condition or exact. In contrast, the *total error* ($\varepsilon_{\triangle}t$) at $t = t_{n+1}$ is the actual error accumulated from $t = 0$ to $t = t_{n+1}$ with y_0 as the initial condition. The *local error* may be either positive or negative while the *total error* which accumulates the *local error*, may or may not grow with time. A discretizing algorithm whose total error does not get amplified is considered numerically stable and if algorithms do not possess this property, they are considered numerically unstable.

2.3.2 Errors in State-Space Approach

In the state-space approach, the discretization error mainly depends on the approximation of $e^{\mathbf{A}\Delta t}$ as it appears in several terms of the equation [66]. $e^{\mathbf{A}\Delta t}$ which is also known as the state transition matrix for a time-invariant linear system can be calculated by the approximation series,

$$e^{\mathbf{A}\Delta t} \cong 1 + \mathbf{A}\Delta t + \frac{1}{2!}(\mathbf{A}\Delta t)^2 + \cdots + \frac{1}{n!}(\mathbf{A}\Delta t)^n + \cdots \tag{2.50}$$

Let us approximate this series by the first $K+1$ terms of (2.50):

$$e^{\mathbf{A}\Delta t} \cong 1 + \mathbf{A}\Delta t + \frac{1}{2!}(\mathbf{A}\Delta t)^2 + \cdots + \frac{1}{K!}(\mathbf{A}\Delta t)^K \triangleq \mathbf{M} = [m_{ij}] \tag{2.51}$$

Then the error matrix $\mathbf{R}$ can be given by the difference between (2.50) and (2.51);

$$\mathbf{R} = \frac{1}{(K+1)!}(\mathbf{A}\Delta t)^{K+1} + \frac{1}{(K+2)!}(\mathbf{A}\Delta t)^{K+2} + \cdots = \sum_{k=K+1}^{\infty} \frac{(\mathbf{A}\Delta t)^k}{k!} \tag{2.52}$$

Let us consider that $\| \mathbf{A} \|$ be one of the following norms of the square matrix $\mathbf{A}$ of order n i.e.,

$$\| \mathbf{A} \| \triangleq \overset{max}{i} \sum_{j=1}^{n} | a_{ij} | \tag{2.53}$$

or

$$\| \mathbf{A} \| \triangleq \overset{max}{j} \sum_{i=1}^{n} | a_{ij} | \tag{2.54}$$

The norm $\| \mathbf{A} \|$ given by (2.53) is the maximum among the n values, each of which is the sum of $| a_{ij} |$ in the same row and (2.54) is the same except that "row" replaced by "column". As $| a_{ij} | \leq \| \mathbf{A} \|$ for all i and j, it can be proved that [66],

$$\| \mathbf{AB} \| \leq \| \mathbf{A} \| \cdot \| \mathbf{B} \| \tag{2.55}$$

Now applying (2.55) to (2.52), we can get the elements of $\mathbf{R} = [r_{ij}]$;

$$| r_{ij} | \leq \sum_{k=K+1}^{\infty} \frac{1}{k!} \mid (i,j)\ element\ of\ (A\Delta t)^k \mid \tag{2.56}$$

$$\leq \sum_{k=K+1}^{\infty} \frac{1}{k!} \| \mathbf{A}\Delta t \|^k \tag{2.57}$$

$$\leq \frac{\| \mathbf{A}\Delta t \|^{(K+1)}}{(K+1)!} \left[1 + \| A\Delta t \| + \| A\Delta t \|^2 + \cdots\right] \tag{2.58}$$

If $\| A\Delta t \| \leq 1$, the above equation can be written as;

$$| r_{ij} | \leq \frac{\| \mathbf{A}\Delta t \|^{(K+1)}}{(K+1)!} \frac{1}{1 - \| A\Delta t \|} \triangleq U \tag{2.59}$$

Equation (2.59) establishes an upper bound of the error when $\mathbf{M}$ given in (2.51) is used to approximate $e^{\mathbf{A}\Delta t}$. Once U is established, the accuracy of $\mathbf{M}$ becomes evident by comparing it with all m_{ij}. The higher the values of K is used, the more accurate results will be found but at a cost of higher computation time.

2.4 Stability of SIT and RIT Algorithms

As the simulation involves transformation into Z-domain from continuous time domain, using either step-invariant or ramp-invariant transformation, stability of the system must be retained while the transformation is being carried out. A discrete-time system is considered to be BIBO (Bounded Input and Bounded Output) stable if every bounded input sequence excites a bounded output sequence. The primary condition for a discrete-time system with a transfer function $G(z)$ to be stable is that every pole of $G(z)$ must lie inside the unit circle of the Z-plane or have a magnitude of less than one [67]. It can be proven that if the system itself is stable in s-domain, it will remain stable when it is converted into z-domain using either step-invariant or ramp invariant Transformation.

The response of $G(z)$ as $k \to \infty$ is called the ***steady-state*** response of the system. For a stable system, the steady-state response of a step sequence input will be a step sequence output and the steady-state response to a ramp sequence input will be a ramp sequence output. The magnitude of the responses may not be the same for each sequential input. If a system with discrete-time transfer function $G(z)$ is fed with a step sequence input of magnitude a, i.e., $u(k\Delta t) = a$, for $k = 0, 1, 2 \cdots$, then $U(z) = \frac{az}{z-1}$ and the output $y(k\Delta t)$ can be obtained from;

$$Y(z) = G(z)U(z) = G(z)\,\frac{az}{z-1} \tag{2.60}$$

The $Y(z)$ can be written as;

$$Y(z) = \frac{azG(z)}{z-1} = aG(1)\frac{z}{z-1} + \textit{terms due to poles of } G(z) \tag{2.61}$$

If every pole of $G(z)$ lies inside the unit circle of the z-plane, $G(z)$ is stable and its time responses will approach zero as $k \to \infty$ and the steady-state response is

$$y_{ss}(k\Delta t) = \lim_{k\to\infty} y(k\Delta t) = aG(1)1^k = aG(1) \tag{2.62}$$

which is a constant. This indicates that the steady-state response of a system with transfer function $G(z)$ due to a unit-step sequence is equal to $G(1)$ which is a stable quantity.

For an input of $u(k\triangle t) = ak\triangle t$, for $k = 0, 1, 2 \cdots$, a ramp sequence input, $U(z) = \frac{a\triangle tz}{(z-1)^2}$ and

$$Y(z) = G(z)U(z) = G(z)\,\frac{a\triangle tz}{(z-1)^2} \tag{2.63}$$

Which can be written as,

$$Y(z) = \frac{a\triangle tzG(z)}{(z-1)^2} = aG(1)\frac{\triangle tz}{(z-1)^2} + a\triangle tG'(1)\frac{z}{z-1} + terms\ due\ to\ poles\ of\ G(z) \tag{2.64}$$

where, $G'(1) = \frac{dG(z)}{dz}|_{z=1}$. If $G(z)$ is stable, then

$$y_{ss}(k\triangle t) = aG(1)k\triangle t + a\triangle tG'(1) \tag{2.65}$$

which is a constant and the system is stable.

For systems expressed in the state-space form,

$$\mathbf{x}(k+1)\triangle t = \mathbf{A}_d\mathbf{x}(k\triangle t) \tag{2.66}$$

according to Lyapunov stability theorem, if all the eigenvalues of $\mathbf{A}_d$ lie inside the unit circle, then the system is said to be stable. Some observations are made regarding the change in controllability and observability due to the discretization [72]:

- In the process of discretization, the sampling frequency is considered pathological (relative to A_c), if A_c has two eigenvalues with equal real parts and imaginary parts that differ by an integer multiple of the sampling frequency. Otherwise the sampling frequency is considered non-pathological. If the sampling frequency is non-pathological, then ($\mathbf{A_d}$, $\mathbf{B_d}$) are controllable if ($\mathbf{A_c}$, $\mathbf{B_c}$) are controllable, and ($\mathbf{C_d}$, $\mathbf{A_d}$) is observable if ($\mathbf{C_c}$, $\mathbf{A_c}$) is observable. This means in case of pathological sampling, discretization may loose controllability and observability.

- If ($\mathbf{A_c}$, $\mathbf{B_c}$) are not controllable neither are ($\mathbf{A_d}$, $\mathbf{B_d}$) which means controllability cannot be gained through dicretization. Same is true for observability.

- Similarly, stability is not lost by the discretization as long as the eigenvalues lie inside the unit circle.

2.5 Illustrative Examples

Case studies have been performed to demonstrate the use of step-invariant and ramp-invariant transformation in digital simulation and their accuracy, stability and efficiency with respect to other commonly used methods such as Forward Euler (FE) and Trapezoidal (TR) are compared. Both the transfer function approach and the state-space approach have been used for the study.

2.5.1 Transfer Function Approach

In this approach, individual components or branch of components are discretized using step-invariant transformation and their Norton's Equivalent current source model has been developed.The equivalents are then used to solve the circuit using Nodal solution technique. For the sake of simplicity and better understanding, a simple RL network as shown in Fig. 2.5 is considered for which, the linear dynamics can be expressed as:

$$v(t) = L\frac{di(t)}{dt} + Ri(t) \tag{2.67}$$

Laplace transform of the above equation gives;

$$G(s) = \frac{I(s)}{V(s)} = \frac{1}{R+sL} \tag{2.68}$$

Using SIT, the z-domain transfer function of (2.68) can be found as;

$$\begin{aligned}
G(z) &= \frac{I(z)}{V(z)} = \frac{z-1}{z}\mathcal{Z}\left[\frac{G(s)}{s}\right] \quad \Rightarrow \frac{z-1}{z}\mathcal{Z}\left[\frac{1}{s(R+sL)}\right] \\
\frac{I(z)}{V(z)} &= \frac{z-1}{z}\left[\frac{1}{R}\frac{z}{z-1} - \frac{1}{R}\frac{z}{z-e^{-\frac{R\Delta t}{L}}}\right] = \frac{1}{R}\left[\frac{1-e^{-\frac{R\Delta t}{L}}}{z-e^{-\frac{R\Delta t}{L}}}\right] \\
I(z) &= \frac{1}{R}\left(1-e^{-\frac{R\Delta t}{L}}\right)z^{-1}V(z) + e^{-\frac{R\Delta t}{L}}z^{-1}I(z) \qquad (2.69)
\end{aligned}$$

Converting 2.69 into time domain,

$$i(k\Delta t) = \frac{1}{R}\left(1-e^{-\frac{R\Delta t}{L}}\right)v[(k-1)\Delta t] + e^{-\frac{R\Delta t}{L}}i[(k-1)\Delta t] \tag{2.70}$$

This is an explicit algorithm where the value of output (current) depends only on the history value. This algorithm produces a time-delay of one-step as will be seen later,

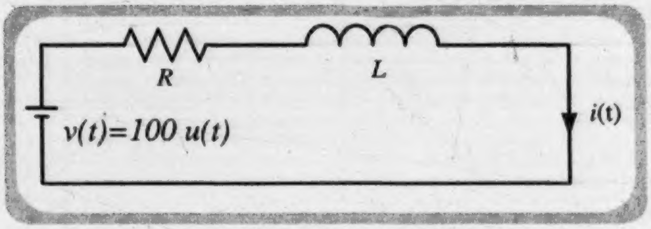


Figure 2.5: An RL network excited by a step-function.

however, using the time-shifted step-invariant transformation (TSSIT), the algorithm becomes implicit and is given as:

$$i(k\Delta t) = \frac{1}{R}\left(1 - e^{-\frac{R\Delta t}{L}}\right) v(k\Delta t) + e^{-\frac{R\Delta t}{L}} i[(k-1)\Delta t] \tag{2.71}$$

Equation (2.71) can be compared with the Norton Equivalent used by the Trapezoidal Rule. As shown in Fig. 2.6, both Trapezoidal Rule and TSSIT algorithm can be expressed through a Norton Equivalent that yields a history term as well as an impedance branch excited by the applied voltage. The history term for TSSIT depends only on the current of the previous time-step, whereas for Trapezoidal Rule, it depends on both the applied voltage and the current of the previous time-step. Similarly, using ramp-invariant transformation,

$$\begin{aligned} G(z) &= \frac{I(z)}{V(z)} = \frac{(z-1)^2}{\Delta t z}\mathcal{Z}\left[\frac{G(s)}{s^2}\right] \quad \Rightarrow \frac{(z-1)^2}{\Delta t z}\mathcal{Z}\left[\frac{1}{s^2(R+sL)}\right] & (2.72)\\ &= \frac{(z-1)^2}{\Delta t z}\mathcal{Z}\left[-\frac{L}{R^2}\frac{1}{s} + \frac{1}{R}\frac{1}{s^2} + \frac{L}{R^2}\frac{1}{(s+R/L)}\right] & (2.73)\\ \frac{I(z)}{V(z)} &= \frac{(z-1)^2}{\Delta t z}\left[-\frac{L}{R^2}\frac{z}{z-1} + \frac{1}{R}\frac{\Delta t z}{(z-1)^2} + \frac{L}{R^2}\frac{z}{(z-e^{-\frac{R\Delta t}{L}})}\right] & (2.74)\\ &= \frac{\left(R\Delta t - L + Le^{-\frac{R\Delta t}{L}}\right) + \left(L - Le^{-\frac{R\Delta t}{L}} - R\Delta t e^{-\frac{R\Delta t}{L}}\right) z^{-1}}{R^2\Delta t\left(1 - e^{-\frac{R\Delta t}{L}} z^{-1}\right)} & (2.75) \end{aligned}$$

Converting it into time domain, we get:

$$\begin{aligned} i(k\Delta t) &= \frac{\left(R\Delta t - L + Le^{-\frac{R\Delta t}{L}}\right)}{R^2\Delta t} v(k\Delta t) + \frac{\left(L - Le^{-\frac{R\Delta t}{L}} - R\Delta t e^{-\frac{R\Delta t}{L}}\right)}{R^2\Delta t} v[(k-1)\Delta t] \\ &+ e^{-\frac{R\Delta t}{L}} i[(k-1)\Delta t] \end{aligned} \tag{2.76}$$

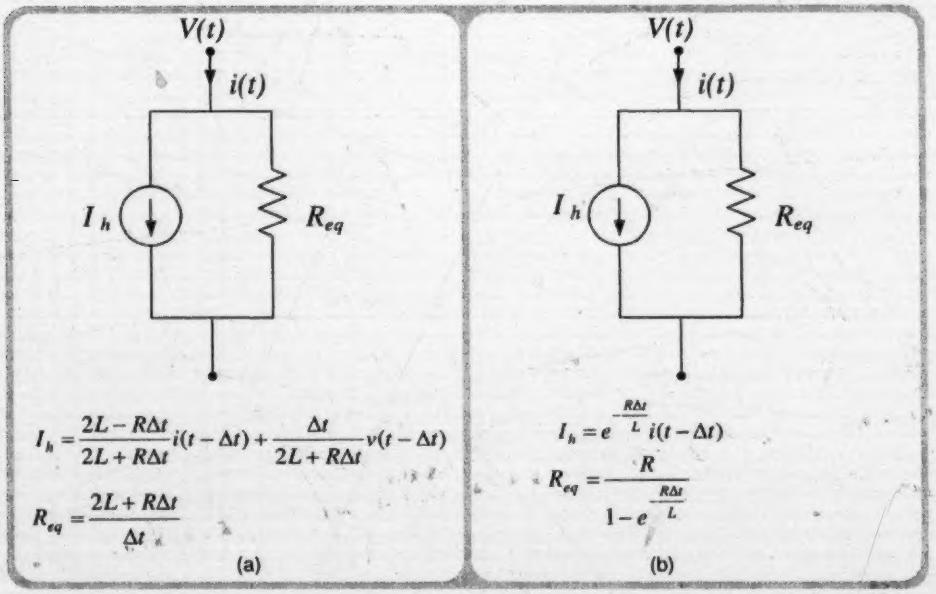

$V(t)$
$i(t)$
I_h
R_{eq}
$I_h = \frac{2L - R\Delta t}{2L + R\Delta t} i(t - \Delta t) + \frac{\Delta t}{2L + R\Delta t} v(t - \Delta t)$
$R_{eq} = \frac{2L + R\Delta t}{\Delta t}$
(a)
$V(t)$
$i(t)$
I_h
R_{eq}
$I_h = e^{-\frac{R\Delta t}{L}} i(t - \Delta t)$
$R_{eq} = \frac{R}{1 - e^{-\frac{R\Delta t}{L}}}$
(b)

Figure 2.6: Norton Equivalent of RL network (a) Trapezoidal rule (b) TSSIT.

Most of the circuits are formed by resistor, capacitor and inductor or their combinations. Norton Equivalents using TSSIT can be derived for all cases except a lonesome capacitor. As the single capacitor does not have any pole, it is not possible to derive Norton Equivalent for it, instead Trapezoidal rule is used in such a case. In that sense, TSSIT can be used as a complementary tool to Trapezoidal rule and a combination of the two can give better accuracy and ease of computation for real-time simulation. Tables 2.1, 2.2, and 2.3 give the summary of Norton equivalent terms for R, L, C, RL, RC, LC and RLC circuits using the SIT, TSSIT and RIT, respectively. Detail derivations are given in Appendix A.

2.5.1.1 Comparison of Accuracy of SIT, TSSIT and RIT

To study and compare the accuracy of the SIT, TSSIT and RIT, the RL circuit of Fig. 2.5 has been chosen. The circuit is excited by a step-function and a sinusoidal voltage sources.

Table 2.1: Norton equivalents for circuit elements using SIT.

Elements	SIT
R	$R_{eq} = R$ $I_h = 0$
L	$R_{eq} = 0$ $I_h = \frac{\Delta t}{L} v[(k-1)\Delta t] + i[(k-1)\Delta t]$
C	Not Possible (Please see Appendix A)
RL	$R_{eq} = 0$ $I_h = \frac{1}{R}\left(1 - e^{-\frac{R\Delta t}{L}}\right) v[(k-1)\Delta t] + e^{-\frac{R\Delta t}{L}} i[(k-1)\Delta t]$
RC	$R_{eq} = R$ $I_h = e^{-\frac{\Delta t}{RC}} i[(k-1)\Delta t] - \frac{1}{R} v(k-1)\Delta t$
LC	$R_{eq} = 0$ $I_h = \gamma \sin \omega \Delta t v[(k-1)\Delta t] - \gamma \sin \omega \Delta t v[(k-2)\Delta t]$ $+2\cos \omega \Delta t i[(k-1)\Delta t] - i[(k-2)\Delta t]$ $\gamma = \sqrt{\frac{C}{L}}$ and $\omega = \sqrt{\frac{1}{LC}}$
RLC	$R_{eq} = 0$ $I_h = \frac{e^{-\alpha\Delta t} - e^{-\beta\Delta t}}{L(\beta-\alpha)} [v(k-1)\Delta t - v(k-2)\Delta t] + (e^{-\alpha\Delta t} + e^{-\beta\Delta t}) i(k-1)\Delta t$ $-e^{-(\alpha+\beta)\Delta t} i(k-2)\Delta t$, where $\alpha = \frac{-\frac{R}{L} + \sqrt{(\frac{R}{L})^2 - \frac{4}{LC}}}{2}$ and $\beta = \frac{-\frac{R}{L} - \sqrt{(\frac{R}{L})^2 - \frac{4}{LC}}}{2}$

Table 2.2: Norton equivalents for circuit elements using TSSIT.

Elements	TSSIT
R	$R_{eq} = R$ $I_h = 0$
L	$R_{eq} = \frac{L}{\Delta t}$ $I_h = i[(k-1)\Delta t]$
C	Not Possible
RL	$R_{eq} = \frac{R}{\left(1 - e^{-\frac{R\Delta t}{L}}\right)}$ $I_h = e^{-\frac{R\Delta t}{L}} i[(k-1)\Delta t]$
RC	Not applicable
LC	$R_{eq} = \gamma \sin \omega \Delta t$ $I_h = -\gamma \sin \omega \Delta t v[(k-1)\Delta t] + 2\cos \omega \Delta t i[(k-1)\Delta t] - i[(k-2)\Delta t]$
RLC	$R_{eq} = \frac{e^{-\alpha\Delta t} - e^{-\beta\Delta t}}{L(\beta-\alpha)}$ $I_h = \frac{e^{-\alpha\Delta t} - e^{-\beta\Delta t}}{L(\beta-\alpha)} [-v(k-2)\Delta t] + (e^{-\alpha\Delta t} + e^{-\beta\Delta t}) i(k-1)\Delta t$ $-e^{-(\alpha+\beta)\Delta t} i(k-2)\Delta t$

Table 2.3: Norton equivalents for circuit elements using RIT.

Elements	RIT
R	$R_{eq} = R$ $I_h = 0$
L	$R_{eq} = \frac{2L}{\Delta t}$ $I_h = i[(k-1)\Delta t] + \frac{\Delta t}{2L} v[(k-1)\Delta t]$
C	Not Possible
RL	$R_{eq} = \frac{R^2 \Delta t}{\left(R\Delta t - L + Le^{-\frac{R\Delta t}{L}}\right)}$ $I_h = \frac{\left(L - Le^{-\frac{R\Delta t}{L}} - R\Delta t e^{-\frac{R\Delta t}{L}}\right)}{R^2 \Delta t} v[(k-1)\Delta t] + e^{-\frac{R\Delta t}{L}} i[(k-1)\Delta t]$
RC	$R_{eq} = \frac{C(1-e^{-\frac{\Delta t}{RC}})}{\Delta t}$ $I_h = e^{-\frac{\Delta t}{RC}} i[(k-1)\Delta t] - \frac{C}{\Delta t}(1 - e^{-\frac{\Delta t}{RC}})[v(k-1)\Delta t]$

Case I: Transient Response with Step-Function Excitation

The first case involves the simulation of an RL circuit excited through a step-function with a magnitude of $100V$ as shown in Fig. 2.5. The RL network is chosen as it is commonly used by many researchers to validate the numerical algorithms through a step-response [34, 69]. A value of $R = 1\Omega$ and $L = 0.05mH$ has been chosen which gives a time constant $\tau = 50\mu s$.

The network is linear and simple, whose exact response with time is given as:

$$i_{ex} = i(t) = 100\left(1 - e^{-200t}\right) \tag{2.77}$$

The response of the network of Fig. 2.5 using Forward Euler (FE), Trapezoidal Rule

Table 2.4: Discretization errors produced by different methods for $\Delta t = \tau = 50\mu s$.

Time	I-exact	Forward Euler			Trapezoidal Rule		
(μs)	Amps	I-FE	ε_T	$\%\varepsilon_T$	I-TR	ε_T	$\%\varepsilon_T$
50	63.212056	0.000	63.212056	100.00	33.333333	29.87872	47.26
100	86.466472	100.0	-13.533528	15.65	77.777778	8.688694	10.04
150	95.021293	100.0	-4.978707	5.24	92.592593	2.428700	2.55
200	98.168436	100.0	-1.831564	1.86	97.530864	0.637572	0.648
250	99.326205	100.0	-0.673795	0.67	99.176955	0.149250	0.015
300	99.752125	100.0	-0.247875	0.24	99.725652	0.026473	0.002
350	99.908812	100.0	-0.091188	0.09	99.908551	0.000261	0.000
400	99.966454	100.0	-0.033546	0.03	99.969517	-0.00306	0.000

(TR), SIT, TSSIT and RIT are given as:

$$i_{FE} = i(k\Delta t) = \left(1 - \frac{R\Delta t}{L}\right) i(k-1)\Delta t + \frac{\Delta t}{L} v(k-1)\Delta t \tag{2.78}$$

$$i_{TR} = i(k\Delta t) = \frac{2 - \frac{R\Delta t}{L}}{2 + \frac{R\Delta t}{L}} i(k-1)\Delta t + \frac{\frac{\Delta t}{L}}{2 + \frac{R\Delta t}{L}} [v(k-1)\Delta t + v(k\Delta t)] \tag{2.79}$$

$$i_{SIT} = i(k\Delta t) = \frac{1}{R}\left(1 - e^{-\frac{R\Delta t}{L}}\right) v[(k-1)\Delta t] + e^{-\frac{R\Delta t}{L}} i[(k-1)\Delta t] \tag{2.80}$$

$$i_{TSSIT} = i(k\Delta t) = \frac{1}{R}\left(1 - e^{-\frac{R\Delta t}{L}}\right) v(k\Delta t) + e^{-\frac{R\Delta t}{L}} i[(k-1)\Delta t] \tag{2.81}$$

$$\begin{aligned} i_{RIT} &= i(k\Delta t) = \frac{\left(R\Delta t - L + Le^{-\frac{R\Delta t}{L}}\right)}{R^2\Delta t} v(k\Delta t) \qquad (2.82) \\ &+ \frac{\left(L - Le^{-\frac{R\Delta t}{L}} - R\Delta t e^{-\frac{R\Delta t}{L}}\right)}{R^2\Delta t} v[(k-1)\Delta t] + e^{-\frac{R\Delta t}{L}} i[(k-1)\Delta t] \end{aligned}$$

A time-step of $\Delta t = \tau = 50\mu s$ has been chosen and the responses produced by different methods are shown in Fig. 2.7. It is obvious from the figure that simulation results obtained through TSSIT matches with the exact solution showing absolutely no error, whereas others show errors until the system reaches to steady-state. The response produced by the step-invariant (SIT) is similar to TSSIT response except that SIT creates one time-step delay in the output.

Table 2.4 and 2.5 shows the total errors produced by each method with a time-step of $\Delta t = 50\mu s$. From these two tables, it is clear that all the methods except TSSIT produce errors and their outputs finally settles to a steady-state value after some time.

For further study, the time-step of the simulation has been increased to $\Delta t =$

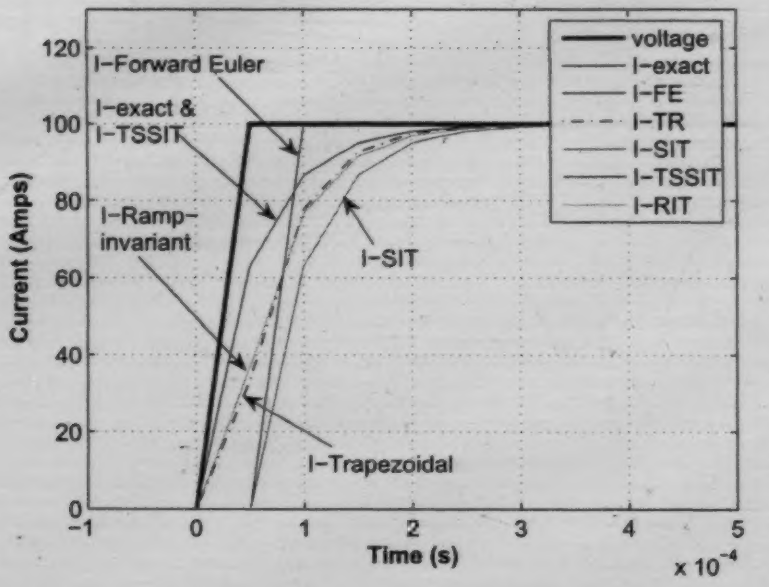


Figure 2.7: Response of the RL network excited by a step-function, using different discretization methods with a time-step $\Delta t = \tau = 50\mu s$.

$2\tau = 100\mu s$, FE becomes oscillatory and beyond $\Delta t = 100\mu s$, it becomes unstable. Therefore, the FE method has been excluded and further study of the response has been carried out at a time-step of $\Delta t = 5\tau = 250\mu s$. The errors are given in Table 2.6 and Table 2.7 and the response are plotted in Fig. 2.8.

Fig. 2.8 indicates that at $\Delta t = 5\tau = 250\mu s$, the TSSIT response matches with the exact response whereas SIT shows one time-step delay. Trapezoidal Rule shows oscillation which decays slowly and finally settles to a steady-state value. However, the RIT method produces a better response than the Trapezoidal response and settles to steady-state value faster without any oscillation. It can be concluded that out of the four methods, the Trapezoidal rule produces maximum error as well as oscillation in the response at this time-step. Another important point to note is that the three methods (SIT, TSSIT and RIT) can be used at a much larger time-step than the time-constant of the network.

Table 2.5: Discretization errors produced by different methods for $\Delta t = \tau = 50\mu s$.(contd.)

Time	TSSIT		Step-Invariant			Ramp-Invariant		
(μs)	I-TSSIT	ε_T	I-SIT	ε_T	% ε_T	I-RIT	ε_T	% ε_T
50	63.212056	0	0.00000	63.212056	100.00	36.787944	26.424112	41.8
100	86.466472	0	63.212056	23.254416	26.893	76.745584	9.720888	11.24
150	95.021293	0	86.466472	8.554821	9.003	91.445179	3.576114	3.76
200	98.168436	0	95.021293	3.147143	3.205	96.852857	1.315579	1.340
250	99.326205	0	98.168436	1.157769	1.165	98.842231	0.483974	0.487
300	99.752125	0	99.326205	0.425920	0.426	99.574081	0.178044	0.178
350	99.908812	0	99.752125	0.156687	0.156	99.843313	0.065499	0.065
400	99.966454	0	99.908812	0.057642	0.057	99.942358	0.024096	0.024

Table 2.6: Discretization errors produced by different methods for $\Delta t = 5\tau = 250\mu s$.

Time	I-exact	Trapezoidal Rule			Step-Invariant		
(μs)	Amps	I-TR	ε_T	% ε_T	I-SIT	ε_T	%ε_T
250	99.32620	71.42857	27.8976	28.088	0.000000	99.3262	100.0
500	99.99546	112.2448	-12.2494	-12.251	99.32620	0.66925	0.669
750	99.99996	94.75218	5.24778	5.2477	99.99546	0.00450	0.004
1000	100.0000	102.2490	-2.24906	-2.249063	100.0000	0.00000	0.000
1250	100.0000	99.03611	0.96388	0.963884	100.0000	0.00000	0.000
1500	100.0000	100.4130	-0.41309	-0.413093	100.0000	0.00000	0.000
1750	100.0000	99.82296	0.17704	0.17704	100.0000	0.00000	0.000
2000	100.0000	100.07587	-0.07587	-0.075874	100.0000	0.00000	0.000

Table 2.7: Discretization errors produced by different methods for $\Delta t = 5\tau = 250\mu s$.(contd.)

Time	I-exact	TSSIT			Ramp-Invariant		
(μs)	Amps	3 I-TSSIT	ε_T	% ε_T	I-RIT	ε_T	% ε_T
250	99.326205	99.326205	0.000	0.000	80.134759	19.191446	19.321
500	99.995460	99.995460	0.000	0.000	99.866149	0.129311	0.1293
750	99.999969	99.999969	0.000	0.000	99.999098	0.000871	0.0008
1000	100.0000	100.000000	0.000	0.000	99.999994	0.000006	0.0006
1250	100.0000	100.000000	0.000	0.000	100.000000	0.000	0.000
1500	100.0000	100.000000	0.000	0.000	100.0000	0.000	0.000
1750	100.0000	100.000000	0.000	0.000	100.0000	0.000	0.000

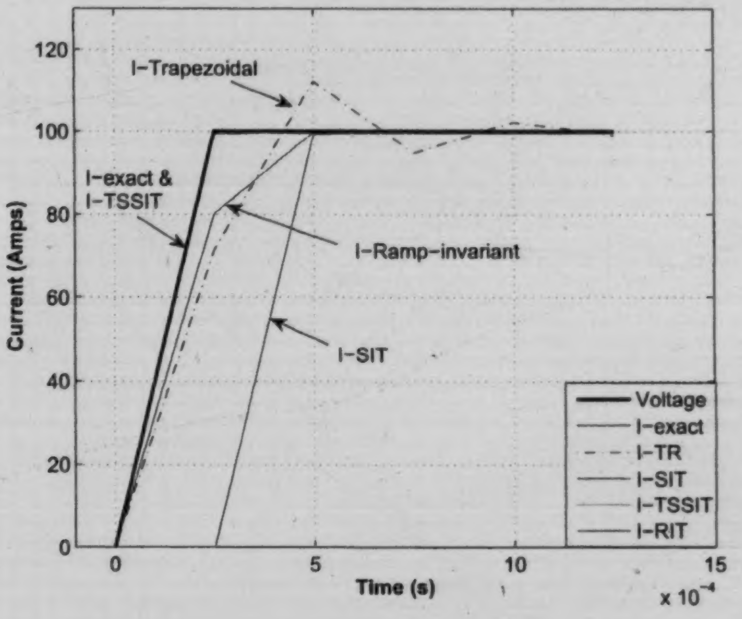


Figure 2.8: Response of the RL network, excited by a step-function, using different discretization methods with a time-step $\Delta t = 5\tau = 250\mu s$.

Case II: Steady-State Response with Sinusoidal Excitation

In this case, the same RL circuit of Fig. 2.5 is now excited by a sinusoidal voltage with a frequency of 2 kHz. A time-step of $\Delta t = 10\mu s$ has been used to simulate the network using all the five methods. The responses obtained through all these methods are shown in Fig. 2.9(a). It has been found that for a sinusoidal voltage input, the steady-state response of the network obtained through Trapezoidal and RIT closely matches with the exact response. The rest of the methods show larger errors. The zoomed view of the responses is given in Fig. 2.9(b). Table 2.8 lists the errors in detail for the two closely matched methods (Trapezoidal and RIT).

Table 2.8 shows that at the very beginning of the simulation, the Trapezoidal Rule shows larger error than the RIT. However, this is not always true and after a while the Trapezoidal Rule has been found with smaller error than RIT. In fact, there is an

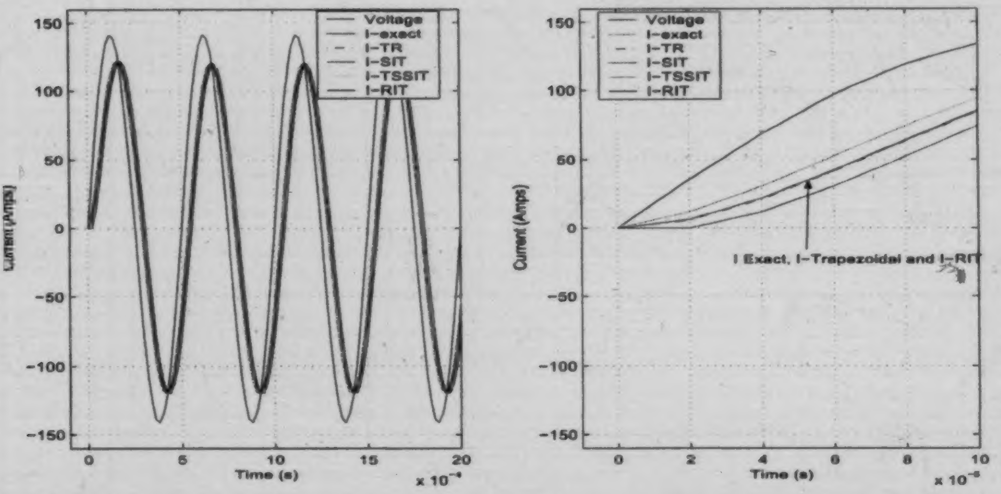


Figure 2.9: Steady-state response of the RL network when excited by a sinusoidal voltage source and simulated with a time-step $\Delta t = 10\mu s$.

Table 2.8: Discretization errors produced by different methods for sinusoidal input at $\Delta t = 10\mu s$.

Time	I-exact	Trapezoidal Rule			Ramp-Invariant		
(μs)	Amps	I-TR	$\varepsilon_{\Delta} t$	$\%\varepsilon_T$	I-RI	ε_T	$\%\varepsilon_T$
10	1.661284	1.610532	0.050752	3.055	1.659157	0.002127	0.120
20	6.210820	6.123933	0.086887	1.400	6.202760	0.008060	0.123
30	13.048638	12.936686	0.111952	0.858	13.031628	0.01701	0.130
40	21.634478	21.505854	0.128624	0.594	21.606211	0.028267	0.130
50	31.478945	31.340027	0.138918	0.441	31.437760	0.041185	0.130
60	42.137009	41.992668	0.144341	0.342	42.081829	0.05518	0.130
70	53.203377	53.057364	0.146013	0.274	53.133659	0.069718	0.131
80	64.309354	64.164587	0.144767	0.225	64.225039	0.084315	0.131

alternate trend for both methods to be closer to the exact solution. It must be noted from Table 2.8 that the error is almost negligible and it could be concluded that both methods produce least errors in compare to others.

Case III: Transient Response with High Frequency Sinusoidal Excitation

A transient response has been studied for the same RL circuit when it was excited by a high frequency (2 kHz) sinusoidal signals. The magnitude of the excitation voltage was suddenly changed to zero from its peak value. The response has been shown in Fig. 2.10. It has been found that while the system was in steady state, the

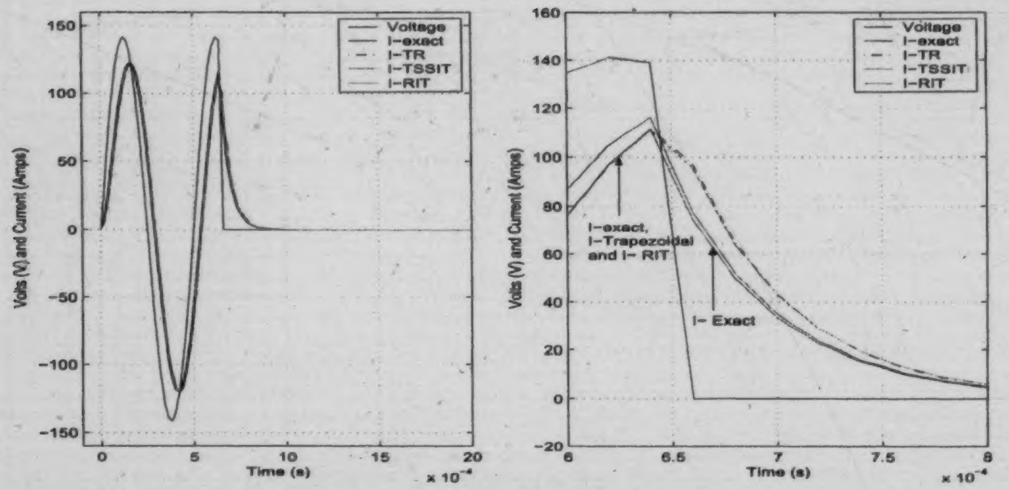


Figure 2.10: Transient response of the RL network when excited by a sinusoidal voltage source and simulated with a time-step $\Delta t = 10\mu s$.

exact response was very close to the response obtained through the Trapezoidal and the RIT methods. However, when a transient situation is created, the exact response took a different path which is closer to the response observed by the TSSIT method.

2.5.2 State-Space Approach

The discrete-time state-space equation for Forward Euler is,

$$\mathbf{x}(k+1)\Delta t = (\mathbf{I} + \mathbf{A_c}\Delta t)\mathbf{x}(k\Delta t) + \mathbf{B_c}\Delta t\mathbf{u}(k\Delta t) \tag{2.83}$$

and using the Trapezoidal Rule, the equation becomes,

$$\begin{aligned}\mathbf{x}(k+1)\Delta t &= \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A_c}\right)^{-1}\left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{A_c}\right)\mathbf{x}(k\Delta t) \\ &+ \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A_c}\right)^{-1}\frac{\Delta t}{2}\mathbf{B_c}[\mathbf{u}(k\Delta t) + \mathbf{u}(k+1)\Delta t]\end{aligned} \tag{2.84}$$

On the other hand, if step-invariant transformation is used, then the state-space equation is,

$$\mathbf{x}(k+1)\Delta t = \mathbf{A_d}\mathbf{x}(k\Delta t) + \mathbf{B_d}\mathbf{u}(k\Delta t) \tag{2.85}$$

$$\mathbf{y}(k\Delta t) = \mathbf{C_d}\mathbf{x}(k\Delta t) + \mathbf{D_d}\mathbf{u}(k\Delta t) \tag{2.86}$$

where, $\mathbf{A}_d = e^{\mathbf{A_c}\Delta t} \cong \mathbf{1} + \mathbf{A}\Delta t + \frac{1}{2!}(\mathbf{A}\Delta t)^2 + \frac{1}{3!}(\mathbf{A}\Delta t)^3 + \frac{1}{4!}(\mathbf{A}\Delta t)^4 \frac{1}{5!}(\mathbf{A}\Delta t)^5 +$ higher order terms (neglected), $\mathbf{B}_d = \left(\int_0^{\Delta} te^{\mathbf{A_c}\tau} d\tau\right) \mathbf{B_c} = [e^{\mathbf{A_c \Delta t}} - \mathbf{1}]\mathbf{A_c}^{-1}\mathbf{B_c} = [1.\Delta t + \frac{1}{2!}\mathbf{A_c}\Delta t^2 + \frac{1}{3!}\mathbf{A_c}^2\Delta t^3 + \frac{1}{4!}\mathbf{A_c}^3\Delta t^4 + \frac{1}{5!}\mathbf{A_c}^4\Delta t^5]\mathbf{B_c} +$ higher order terms (neglected), $\mathbf{C_c} = \mathbf{C_d}$ and $\mathbf{D_c} = \mathbf{D_d}$. The reason for neglecting the higher order terms is the use of very small time-step, Δt (in the range of $50\mu s$ to $100\mu s$). At such a small time-step, the contribution from higher order terms is almost zero.

Similar to the transfer function based approach, in the state-space approach, it has been found that the SIT produces a one time-step delay in the simulation output. However, this delay can be overcome by using the input of the next time-step i.e. using the value of $\mathbf{u}(k+1)\Delta t$ instead of $\mathbf{u}(k\Delta t)$ in equations (2.85), while keeping the discretized system matrices unchanged. The modified version is TSSIT and is given as,

$$\mathbf{x}(k+1)\Delta t = \mathbf{A_d}\mathbf{x}(k\Delta t) + \mathbf{B_d}\mathbf{u}(k+1)\Delta t \tag{2.87}$$

$$\mathbf{y}(k\Delta t) = \mathbf{C_d}\mathbf{x}(k\Delta t) + \mathbf{D_d}\mathbf{u}(k\Delta t) \tag{2.88}$$

Finally, the RIT equations can be written as:

$$\mathbf{x}(k+1)\Delta t = \mathbf{F_d}\mathbf{x}(k\Delta t) + \mathbf{G_d}u(k\Delta t) + \mathbf{H_d}\mathbf{u}(k+1)\Delta t \tag{2.89}$$

$$\mathbf{y}(k\Delta t) = \mathbf{C_d}\mathbf{x}(k\Delta t) + \mathbf{D_d}\mathbf{u}(k\Delta t) \tag{2.90}$$

where,

$$\mathbf{F_d} = e^{\mathbf{A_c}\Delta t} \tag{2.91}$$

$$\begin{aligned}
\mathbf{G_d} &= \sum_{n=0}^{\infty} \frac{1}{n(n+2)!}(\mathbf{A_c}\Delta t)^n \times \mathbf{B_c}\Delta t \\
&= \mathbf{B_c}\frac{\Delta t}{2} + \frac{1}{3}(\mathbf{A_c}\Delta t)(\mathbf{B_c}\Delta t) + \frac{1}{8}(\mathbf{A_c}\Delta t)^2(\mathbf{B_c}\Delta t) + \\
&\quad \frac{1}{30}(\mathbf{A_c}\Delta t)^3(\mathbf{B_c}\Delta t) + \frac{1}{144}(\mathbf{A_c}\Delta t)^4(\mathbf{B_c}\Delta t) + \frac{1}{840}(\mathbf{A_c}\Delta t)^5(\mathbf{B_c}\Delta t)
\end{aligned} \tag{2.92}$$

$$\begin{aligned}
\mathbf{H_d} &= \sum_{n=0}^{\infty} \frac{1}{(n+2)!}(\mathbf{A_c}\Delta t)^n \times \mathbf{B_c}\Delta t \\
&= \mathbf{B_c}\frac{\Delta t}{2} + \frac{1}{6}(\mathbf{A_c}\Delta t)(\mathbf{B_c}\Delta t) + \frac{1}{24}(\mathbf{A_c}\Delta t)^2(\mathbf{B_c}\Delta t) + \\
&\quad \frac{1}{120}(\mathbf{A_c}\Delta t)^3(\mathbf{B_c}\Delta t) + \frac{1}{720}(\mathbf{A_c}\Delta t)^4(\mathbf{B_c}\Delta t) + \\
&\quad \frac{1}{5040}(\mathbf{A_c}\Delta t)^5(\mathbf{B_c}\Delta t)
\end{aligned} \tag{2.93}$$

$$\mathbf{C_d} = \mathbf{C_c} \tag{2.94}$$

$$\mathbf{D}_d = \mathbf{D}_c \tag{2.95}$$

The higher order terms are again excluded as their contribution with smaller time-step is negligible. Besides, their omission reduces the computational burden through the elimination of intensive matrix calculations.

Case I: Transient response with step-function excitation at a time-step $\Delta t = 50\mu$s

A series RLC circuit as shown in Fig. 2.11, has been chosen for using state-space solution technique. The input to the system is a step-function given as,

$$U(t) = 10u(t) \tag{2.96}$$

The parameters of the RLC circuit are chosen arbitrarily, keeping in mind that the step response becomes oscillatory with a decaying amplitude. A value of $R = 1\Omega$, $L = 10mH$ and $C = 25\mu F$ produces the following state-space equation,

$$\dot{x}(t) = \begin{bmatrix} -100 & -100 \\ 40000 & 0 \end{bmatrix} x + \begin{bmatrix} 100 \\ 0 \end{bmatrix} u \tag{2.97}$$

$$y = \begin{bmatrix} 1 & 1 \end{bmatrix} x \tag{2.98}$$

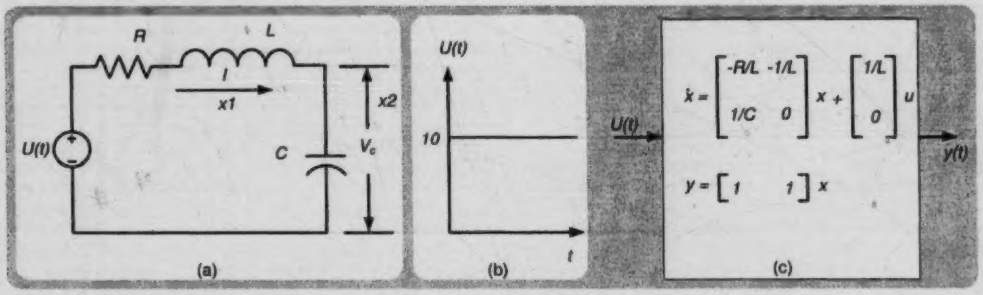

R
L
i
x1
x2
U(t)
C
V_c
U(t)
10
t
U(t)
$\dot{x} = \begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix} x + \begin{bmatrix} 1/L \\ 0 \end{bmatrix} u$
$y = \begin{bmatrix} 1 & 1 \end{bmatrix} x$
y(t)
(a)
(b)
(c)

Figure 2.11: (a) RLC network, (b) step-function input, (c) Equivalent state-space model

where, the system matrices are, $\mathbf{A_c} = \begin{bmatrix} -100 & -100 \\ 40000 & 0 \end{bmatrix}$, $\mathbf{B_c} = \begin{bmatrix} 100 \\ 0 \end{bmatrix}$, $\mathbf{C_c} = \begin{bmatrix} 1 & 1 \end{bmatrix}$ and $D_c = 0$. The above state-space equation can now be solved using different solution methods described at (2.83), (2.84), (2.87) and (2.89).

The results obtained from the simulation are shown in Fig. 2.12. Forward Euler creates instability when the time-step is increased and also shows maximum error in comparison to other methods. Therefore, it has been excluded from the comparison. A Laplace transform based solution known as the *exact solution* is used to compare the results obtained through the other methods.

From Fig. 2.12, it has been found that the solution obtained through the TSSIT method matches accurately with the exact solution. However, the SIT shows the maximum error due to its inherent delay of one time-step. For this particular case also, the RIT has been found to match closely with the Trapezoidal method. Similar results have also been observed when a transfer function based solution technique was used. Similar to the current response, the voltage response has also been found to behave the same way and is shown in Fig. 2.13. Table 2.9 and Table 2.10 reveal the total errors produced by each method.

Case II: Transient Response with Step-Function Excitation at a time-step T=250μs

In this case study, the time-step has been increased to $\Delta t = 250\mu s$ and the same RLC circuit is simulated using different methods for the state-space solution. Results

Table 2.9: Discretization errors produced by different methods for sinusoidal input solved using state-space method at a time-step of $\Delta t = 50\mu$ s

Time	I-exact	Trapezoidal Rule			Step-Invariant		
(μs)	Amps	I-TR	ε_T	% ε_T	I-SI	ε_T	% ε_T
50	99.326205	71.428571	27.897634	28.086	0.000000	99.326205	100.00
100	99.995460	112.244898	-12.249438	-12.249	99.326205	0.669255	0.6737
150	99.999969	94.752187	5.247782	5.247	99.995460	0.004509	0.0045
200	100.0000	102.249063	-2.249063	-2.249	99.999969	0.000031	0.000
250	100.0000	99.036116	0.963884	0.9638	100.0000	0.000000	0.000
300	100.0000	100.413093	-0.413093	-0.413	100.0000	0.000000	0.000
350	100.0000	99.822960	0.17704	0.17704	100.0000	0.000000	0.000
400	100.0000	100.075874	-0.075874	-0.075874	100.0000	0.000000	0.000

Table 2.10: Discretization errors produced by different methods for sinusoidal input solved using state-space method at a time-step of $\Delta t = 50\mu$ s. (contd.)

Time	I-exact	TSSIT			Ramp-Invariant		
(μs)	Amps	I-TSSIT	$\varepsilon_\Delta t$	% $\varepsilon_\Delta t$	I-RI	$\varepsilon_\Delta t$	% $\varepsilon_\Delta t$
50	99.326205	99.326205	0.0	0.0	80.134759	19.191446	19.32
100	99.995460	99.995460	0.0	0.0	99.866149	0.129311	0.129
150	99.999969	99.999969	0.0	0.0	99.999098	0.000871	0.087
200	100.0000	100.000000	0.0	0.0	99.999994	0.000006	0.006
250	100.0000	100.000000	0.0	0.0	100.000000	0.0	0.0
300	100.0000	100.000000	0.0	0.0	100.0000	0.0	0.0
350	100.0000	100.000000	0.0	0.0	100.0000	0.0	0.0
400	100.0000	100.000000	0.0	0.0	100.0000	0.0	0.0

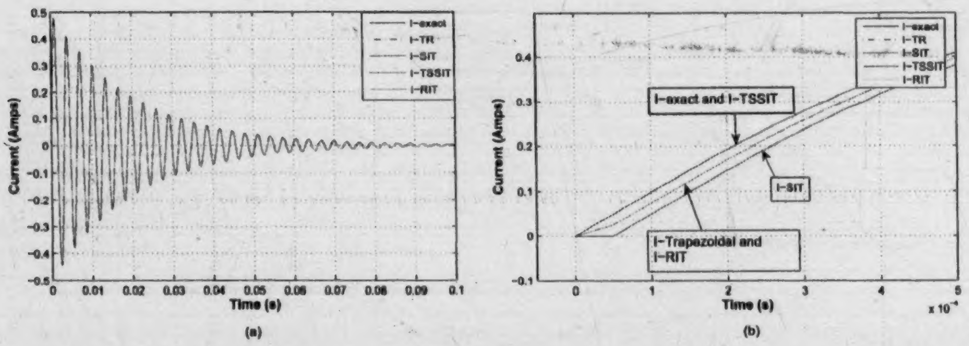


Figure 2.12: (a) Step response of an RLC network solved by state-space solution technique with different discretization methods at a time-step $\Delta t = 50\mu s$, (b) Zoomed view of the response.

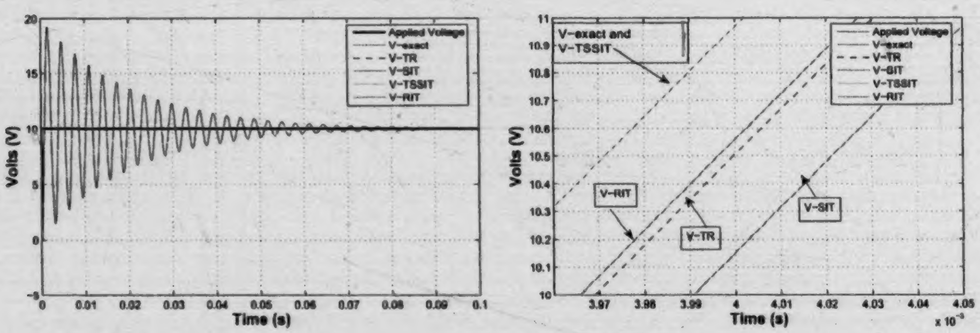


Figure 2.13: (a) Step response of an RLC network solved by state-space solution technique with different discretization methods at a time-step $\Delta t = 50\mu s$, (b) Zoomed view of the response.

are shown in Fig. 2.14 and Fig. 2.15. It was found that even with a time-step of $\Delta t = 250\mu s$, the TSSIT method produces the same response as it is observed in case of the exact solution. The main reason for this is the non-variance of input. However, RIT and SIT produce errors as seen in the previous cases. RIT produces less error than SIT as SIT suffers from inherent time-step delay. It is interesting to note that Trapezoidal Rule shows incremental error with simulation time. At the very beginning it starts close to the RIT response but its compounding error slowly moves it away and eventually, with time a large deviation from the exact solution has been observed. This phenomenon has been previously described in [69].

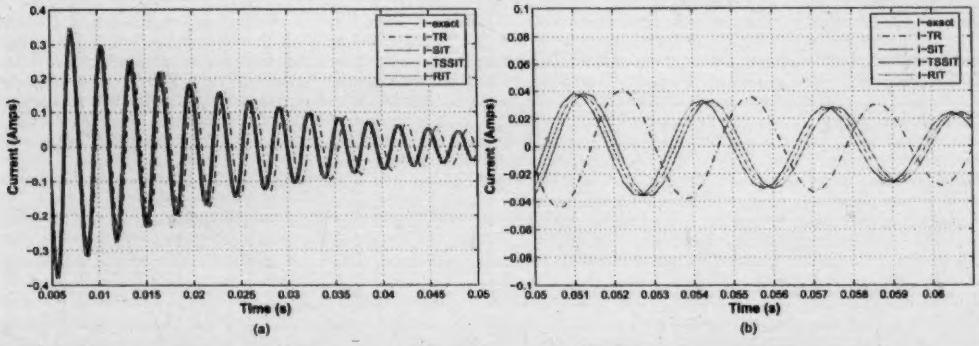


Figure 2.14: (a) Step response of an RLC network solved by state-space solution technique using different discretization methods with a time-step $\Delta t = 250\mu s$, (b) Zoomed view of the response.

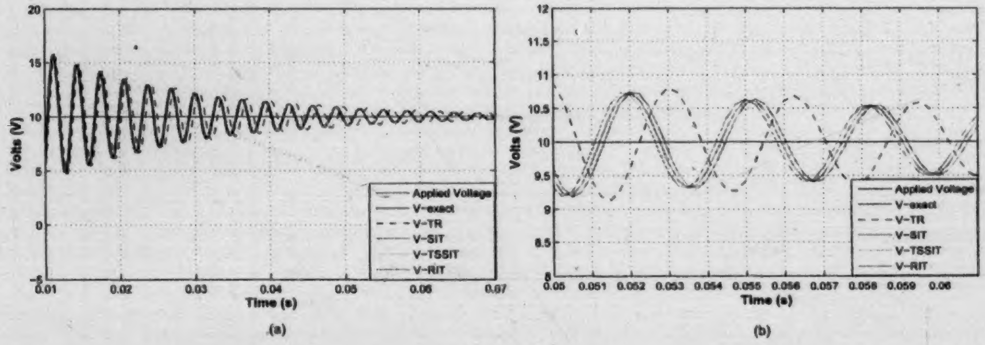


Figure 2.15: (a) Step response of an RLC network solved by state-space solution technique using different discretization methods with a time-step $\Delta t = 250\mu s$, (b) Zoomed view of the response.

Similar behavior has been observed in the voltage response of the RLC network. One important conclusion can be drawn from this study is that SIT, TSSIT and RIT are least affected by the increase in time-step whereas the Forward Euler and the Trapezoidal Rule are greatly affected by the change of time-step.

2.6 Summary

This chapter investigated the accuracy and stability of various discretization techniques which are commonly used in discrete design of digital controllers. The comparative study with the existing discretization techniques has revealed that these techniques can also be used for digital simulation of electrical networks. In the light of the foregone investigation, the following conclusions can be drawn:

- Step-Invariant transformation is an explicit discretizing method which will introduce phase error in the simulation due to its inherent delay. However, the advantage of this method is that it is usable in real-time simulation due to its explicit characteristics.
- Time Shifted Step-Invariant Transformation can be used in case of transients created by sudden change of the inputs.
- Ramp-Invariant and Trapezoidal Rule performs almost similar at small time-step. However, with large time-step, Ramp-Invariant is found to be more stable and accurate than Trapezoidal.
- Considering all these aspects, it is recommended that for off-line digital simulation, Ramp-Invariant can be used for sinusoidal oscillatory inputs such as sine or cosine signal, however, if there is a transient, the algorithm would be replaced by the TSSIT as it performs better in transient situations.
- Despite the fact that both TSSIT and RIT are implicit methods, they can still be used in real-time simulation. If the inputs to the system are predictable, these algorithms can be used for simulation with the predicted inputs and at the beginning of next time-step, an error checking mechanism can be introduced to find out if there is any difference between the actual inputs and the predicted inputs. This will not take much extra time as in many cases, where switching actions take place predominantly, this type of correction algorithms are already in use.

The developed discretization methods are applied in the modeling and simulation of VSC-HVDC system for off-line and HIL simulation in chapter 5. RIT is used for steady-state and TSSIT is used for transients due to faults or sudden changes in inputs.

Chapter 3

Switching Events Synchronization

3.1 Introduction

When a digital controller for a switching power electronic apparatus needs to be tested, the most economic and efficient solution is to perform HIL simulation where the controller is interfaced with the real-time simulator that models the power electronic system and the surrounding power system. The digital controller produces the gating signals which are discrete switching signals to be fed to the real-time simulator. The real-time simulator, which is solving the system differential equations at a certain discrete time-step, $\triangle t$, will therefore require to account the switching events that takes place while the calculation for the time-step is running. These are known as *inter-step* switching events and need proper addressing for an accurate simulation. This chapter[1] concentrates on:

- A family of algorithms with varying level of complexity, for accounting *inter-step* switching events in digital simulation.

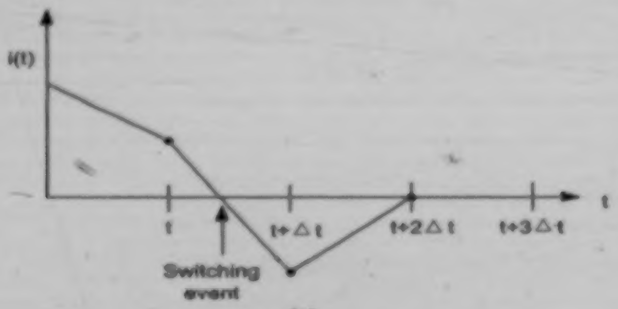
- Off-line simulation results using a Pulse Width Modulated (PWM) VSC.

- A comparative study of their performances such as harmonic contents, errors in fundamental component and simulation time requirement.

---

[1] Materials from this chapter has been published: M. O. Faruque, V. Dinavahi, W. Xu, "Algorithms for the Accounting of Multiple Switching Events in Digital Simulation of Power Electronic Systems", *IEEE Trans. on Power Delivery*, Vol. 20, No. 2, Part-I, pp. 1157 - 1167, April 2005.

The effect of *inter-step* switching events can be explained in Fig. 3.1 where the current in a diode has reduced to zero between t and $t + \Delta t$. Because of the fixed discrete time-step simulation, the information of diode turn-off is available only at time $t + \Delta t$ and the diode current will be accounted as zero at $t + 2\Delta t$.



(a)

Figure 3.1: Current chopping of a diode

In real-time simulation, the use of fixed step faces this challenge when modeling switching circuits. If an *inter-step* switching occurs, it remains unaccounted until the beginning of the next time-step. This results a delay in the instant of switching which brings inaccuracy to the simulation and also produces spikes and numerical oscillation (sometimes called chatter). The number of switching events in any given time-step of the simulator depends on two factors: (1) the switching frequency of the power electronic system i.e. its digital controller sampling rate and (2) the size and complexity of the power electronic system i.e. the number of switches/converters in the system. The higher the switching frequency or the larger the system, the higher is the number of inter-step switching events. Now a days, use of multi-level and multi-pulse converters with high switching frequency is the growing trend in both research and industries. Therefore, multiple switching in one time-step is a common phenomenon. Moreover, the timing of a switching event is not a known priori (since it is controlled by the processes external to the simulator) and it seldom coincides exactly with the beginning of the time-step, Δt.

The inaccuracy due to the *inter-step* switching can be minimized by allowing very small time-step, however, that will require a large simulation time and would be impractical for the implementation of real-time simulation. Various techniques [40] are proposed in the literature to deal with single *inter-step* switching events in digital simulation under both off-line and real-time conditions. One solution for this problem was proposed in [41] is to use a variable time-step so that if switching is detected, the program changes to small time-step and reverts to the original time-step after that. However, a variable time-step requires a re-computation of system matrices at almost every time-step which eventually increases the execution time and real-time simulation might be an impossible task. An alternative way is to use linear interpolation [43,44] to the point of switching and then continuing normal simulation. However, in this technique, the time grid is no more uniformly spaced and to reach to the time grid again, several techniques are proposed in the literature [40].

3.2 Multiple Inter-step Switchings

Even though, correction algorithms for single *inter-step* switching have been available in the literature, solutions to the problem of multiple switching events in one time-step was not found until recently a family of algorithms has been proposed in [28] as an outcome of this research work. Fig. 3.2 illustrates the fixed time-step simulation approach with two switching events occurring within one time-step. $\mathbf{x}_i$, $(i = 0, 1, 2, \ldots)$ represent the states of the system computed by the simulator at every time step Δt and $\mathbf{y}_i$, $(i = 0, 1, 2, \ldots)$ represent the true states of the physical system. In the context of this figure and other figures in this chapter, *Time-Step* refers to the discrete-time interval with which the system differential equations are numerically integrated by the simulator; the term *step-size* is used interchangeably with *time-step*. In contrast, the term *Time-Grid* is chosen to represent the interval at which output data points are generated by the simulator.

In Fig. 3.2, the simulation time-grid is the same as the time-step Δt. The switching events Ⓐ and Ⓑ which occur at times t_A and t_B respectively are accounted at time t_2 when the real-time simulator has already calculated and emitted the incorrect state $\mathbf{x}_2$. Real-time operation does not permit recalling and changing the state $\mathbf{x}_2$. The physical system, on the other hand, would respond to the events with states

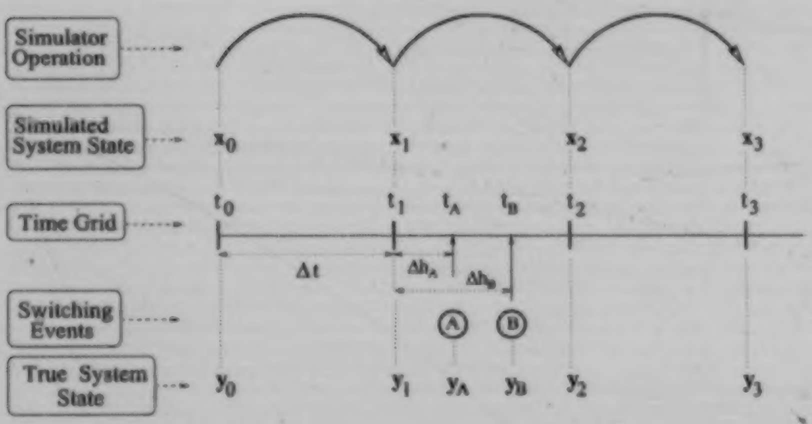


Figure 3.2: Multiple inter-step switching events in digital simulation.

$\mathbf{y}_A$ and $\mathbf{y}_B$ at times t_A and t_B respectively and $\mathbf{y}_2$ would be the true state of the system at time t_2. In a fixed time-step algorithm, at the beginning of every time-step the simulator looks for the switching event and its timing information. With this knowledge it updates the power electronic model in the system. Depending on the type of switches, the circuit variables $\mathcal{V}$, for example voltages at the point of common coupling, may be functions of the firing signals $\mathcal{S}$ and/or the network state $\mathbf{x}$.

$$\mathcal{V} = \mathbf{f}(\mathcal{S}, \mathbf{x}) \tag{3.1}$$

Then the simulator proceeds to obtain the numerical solution of the network state equations for the given time-step.

The state equations for a linear time-invariant network can be written in general as

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u} \tag{3.2}$$

subject to the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$ and $\mathbf{u}(t_0) = \mathbf{u}_0$, where $\mathbf{x} \in \mathbb{R}^N$ is the network state vector and $\mathbf{u} \in \mathbb{R}^M$ is the input vector. $\mathbf{A}$ and $\mathbf{B}$ are constant matrices dependent on the network constants such as R, L, and C. With a time-step of integration $\triangle t$, the solution at time t can be expressed in terms of the solution at time $(t - \triangle t)$:

$$\mathbf{x}(t) = \mathbf{x}(t - \triangle t) + \int_{t-\triangle t}^{t} [\mathbf{A}\,\mathbf{x}(\varsigma) + \mathbf{B}\,\mathbf{u}(\varsigma)]\; d\varsigma \tag{3.3}$$

where ς is a variable of integration. Numerical solution of (3.3) can be obtained by approximating the integral in (3.3) using a numerical integration algorithm. Let the subscript $(n-1)$ denote quantities at time $(t-\Delta t)$ and n quantities at time t. Then,

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \mathbf{g}\left(\Delta t, \mathbf{A}, \mathbf{B}, \mathbf{x}_n, \mathbf{x}_{n-1}, \ldots, \mathbf{u}_n, \mathbf{u}_{n-1}, \ldots\right) \tag{3.4}$$

(3.4) forms the discrete time solution of (3.2). The number of history terms of $\mathbf{x}$ and $\mathbf{u}$ in the integral approximation will depend on the type of the chosen numerical method. Thus, a difference equation can take the form of

$$\mathbf{x}_n = \boldsymbol{\alpha}\,\mathbf{x}_{n-1} + \boldsymbol{\beta}\mathbf{u}_{n-1} + \boldsymbol{\gamma}\mathbf{u}_n \tag{3.5}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are the coefficients whose value depend on the system matrix, time-step and the numerical method used. Therefore, a change in the time-step, Δt would necessitate a re-calculation of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.

3.3 Algorithms for the accounting of multiple switches

As shown in Fig. 3.2, two switching events Ⓐ and Ⓑ arrive in the time interval $(t_2 - t_1)$ while the simulator computes and emits state $\mathbf{x}_2$ at time t_2. Although, real-time operation does not allow changing state $\mathbf{x}_2$, it does allow the simulator to take certain *corrective* action before proceeding to calculate the next state $\mathbf{x}_3$ at time t_3 provided that the simulator has the timing information of the switching events relative to the simulation time-grid. The earliest time at which this information can be made available to the simulator is at the instant an event occurs i.e., t_A and t_B, and the latest is at the end of time t_2. The proposed algorithms described in this section can be broadly classified into two categories:

(1) *Fixed Time-Grid Algorithms*: In these algorithms the output data of the simulation is obtained at equal intervals of time thereby fixing the time-grid. If Δt is varied internally, then a re-synchronization is performed to get the simulation back on the original time-grid. Therefore, these algorithms can have a fixed or a variable simulation time-step. Algorithms I through VII in Fig. 3.3 belong to this category.

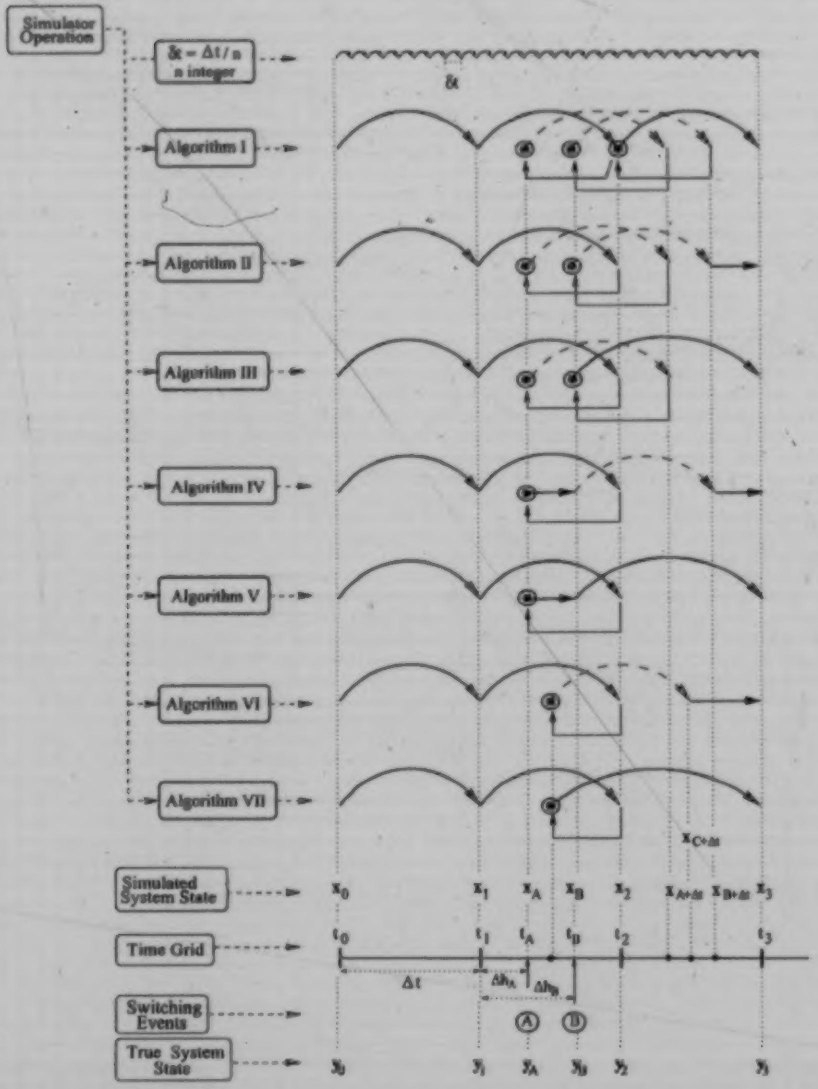


Figure 3.3: Digital simulation algorithms for multiple switching events based on a fixed time-grid. ⌒⌒ : Full-state calculation, ◉ : Linear Interpolation, ⟶ : Linear Extrapolation.

For the sake of clarity these algorithms are illustrated with only two switching events even though their implementation can account a maximum of three events in one time-step. More than three switching events in a time-step seldom happens.

(2) *Variable Time-Grid Algorithms*: In these algorithms, the real-time simulator is

interrupted in its normal operation at the instant a switching event occurs, effectively varying the external simulation time-grid. However, the internal time-step Δt of the simulation is always fixed. Algorithm VIII belongs to this category.

At the beginning of each time-step, the simulator takes one of the two operational paths based on the information it receives about any switching events:

1. Normal Operation is executed in case of no switching event has been detected in the previous time-step.
2. In case of one or more switching events in the previous time step, *Post-Event Correction Operation* is executed. Depending on the algorithm, correction involves linear interpolation, linear extrapolation or a change of time-step. The correction variables include the state variables of the system and the inputs (voltage or current sources). After correction to the point of switching, the system is updated before proceeding to the next time-step. During the updates, inductor currents and capacitor voltages are kept unchanged. The following algorithms explain more details of this mode of operation.

3.3.1 Fixed Time-Grid Algorithms

Once the exact instant of switching event is detected, these algorithms implement post-event corrections. For all the algorithms, the interpolated variables include the state variables, $\mathbf{x}$ (inductor currents and capacitor voltages) and the inputs $\mathbf{u}$ (AC system voltages). These variables are interpolated to the point of switching and then the power electronic converter model (3.1) is updated in accordance with the switching changes. The *updating* procedure depends on the model used to simulate the switches. Using the switching function models, the switch status are flipped and the voltage levels are decided.

Algorithm I

This algorithm is similar to the double interpolation technique that has been used by PSCAD/EMTDC [34]. After calculating state $\mathbf{x}_2$, when the simulator has acknowledged that two events have arrived in the previous simulation interval at times t_A and t_B, the following steps can be taken to *correct* the subsequent state calculation:

1. Based on states $\mathbf{x}_1$ and $\mathbf{x}_2$, state $\mathbf{x}_A^{(1)}$ at time t_A is linearly interpolated as

$$\mathbf{x}_A^{(1)} = \mathbf{x}_1 + \frac{(t_A - t_1)}{\Delta t}(\mathbf{x}_2 - \mathbf{x}_1) \tag{3.6}$$

2. Once the interpolation is performed, the power electronic model is updated according to the switching event Ⓐ. This will yield $\mathbf{x}_A^{(2)}$ which is different from $\mathbf{x}_A^{(1)}$.

3. State $\mathbf{x}_{A+\Delta t}$ at time $(t_A + \Delta t)$ is calculated using (3.5).

4. Now, based on $\mathbf{x}_A^{(2)}$ and $\mathbf{x}_{A+\Delta t}$, state $\mathbf{x}_B^{(1)}$ is interpolated as

$$\mathbf{x}_B^{(1)} = \mathbf{x}_A^{(2)} + \frac{(t_B - t_A)}{\Delta t}\left(\mathbf{x}_{A+\Delta t} - \mathbf{x}_A^{(2)}\right) \tag{3.7}$$

5. The power electronic model (3.1) is again updated using switching event Ⓑ and state $\mathbf{x}_B^{(2)}$ is calculated in accordance with the switching changes at t_B.

6. State $\mathbf{x}_{B+\Delta t}$ at time $(t_B + \Delta t)$ is calculated using (3.5).

7. The corrected state $\widehat{\mathbf{x}}_2$ at time t_2 is now interpolated based on states $\mathbf{x}_B^{(2)}$ and $\mathbf{x}_{B+\Delta t}$ as

$$\widehat{\mathbf{x}}_2 = \mathbf{x}_B^{(2)} + \frac{(t_2 - t_B)}{\Delta t}\left(\mathbf{x}_{B+\Delta t} - \mathbf{x}_B^{(2)}\right) \tag{3.8}$$

   This step puts the simulation on the original time-grid.

8. Finally, state $\mathbf{x}_3$ is calculated from $\widehat{\mathbf{x}}_2$ using (3.5).

There are 3 interpolations and 3 full-state calculations and two model updates involved in going from time t_2 to time t_3, with no change in Δt, after the simulator receives the information about the switching events. Step 7 is solely an internal calculation since it is not possible to change $\mathbf{x}_2$ in real-time. This step may be used in off-line simulations to produce more accurate results.

Algorithm II

After Step 6 in Algorithm I:

7. Using $\mathbf{x}_B^{(2)}$ and $\mathbf{x}_{B+\Delta t}$, state $\mathbf{x}_3$ is linearly extrapolated.

For this step, states $\mathbf{x}_B^{(2)}$ and $\mathbf{x}_{B+\Delta t}$ have been chosen, in particular, because they are fully corrected states. Note that state $\mathbf{x}_2$ is left uncorrected, however, for off-line simulator state $\mathbf{x}_2$ can be corrected to have more accurate results. Thus, in this algorithm there are 2 interpolations, 2 full-state calculations and 1 extrapolation with a fixed $\triangle t$.

Algorithm III

After Step 5 in Algorithm I, i.e., once $\mathbf{x}_B^{(2)}$ has been determined,

6. The step-size of the simulation is changed from $\triangle t$ to $\{\triangle t + (t_2 - t_B)\}$ and $[\boldsymbol{\alpha}\,\boldsymbol{\beta}]_{\{\triangle t+(t_2-t_B)\}}$ is calculated.
7. State $\mathbf{x}_3$ is calculated using (3.5).

Therefore, there are 2 interpolations, 2 full-state calculations and 1 $[\boldsymbol{\alpha}\,\boldsymbol{\beta}]$ re-calculation in this algorithm.

Algorithm IV

After Step 2 in Algorithm I, i.e., after $\mathbf{x}_A^{+}$ has been obtained, this algorithm uses the following steps:

3. $\mathbf{x}_{\mathbf{B}}^{(1)}$ is extrapolated based on $\mathbf{x}_1$ and $\mathbf{x}_A^{(2)}$

$$\mathbf{x}_B^{(1)} = \mathbf{x}_1 + \frac{(t_B - t_1)}{(t_A - t_1)}\left(\mathbf{x}_A^{(2)} - \mathbf{x}_1\right) \qquad (3.9)$$

4. The power electronic model (3.1) is updated to obtain $\mathbf{x}_B^{(2)}$ using switching event Ⓑ.

5. $\mathbf{x}_{B+\Delta t}$ at time $t_B + \Delta t$ is calculated using (3.5).

6. Based on $\mathbf{x}_B^{(2)}$ and $\mathbf{x}_{B+\Delta t}$ state $\mathbf{x}_3$ is extrapolated as

$$\mathbf{x}_3 = \mathbf{x}_B^{(2)} + \frac{(t_3 - t_B)}{\Delta t}\left(\mathbf{x}_{B+\Delta t} - \mathbf{x}_B^{(2)}\right) \tag{3.10}$$

This algorithm is a variation of Algorithm II in which 1 interpolation and 1 full-state calculation has been substituted by one additional extrapolation with Δt remaining fixed.

Algorithm V

After Step 4 in Algorithm IV, i.e., after obtaining $\mathbf{x}_B^{(2)}$ by extrapolation from $\mathbf{x}_1$ and $\mathbf{x}_A^{(2)}$ and updating switching at t_B, this algorithm uses the following steps:

5. The step-size of the simulation is changed from Δt to $\{\Delta t + (t_2 - t_B)\}$ and $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]_{\{\Delta t + (t_2 - t_B)\}}$ is calculated.

6. $\mathbf{x}_3$ is calculated using (3.5).

This algorithm requires 1 interpolation, 1 full state calculation, 1 extrapolation, and 1 $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$ re-calculation. It is a variation of Algorithm III where 1 interpolation and 1 full-state calculation has been replaced by 1 extrapolation.

Algorithm VI

Once the simulator obtains the information of the two switching events then:

1. The system state at $t_C = (t_A + t_B)/2$, the average location of the two events at t_A and t_B, is interpolated based on the states $\mathbf{x}_1$ and $\mathbf{x}_2$.

$$\mathbf{x}_C = \mathbf{x}_1 + \frac{(t_C - t_1)}{\Delta t}(\mathbf{x}_2 - \mathbf{x}_1) \tag{3.11}$$

The power electronic model (3.1) is updated using both switching events Ⓐ and Ⓑ.

2. State $\mathbf{x}_{C+\Delta t}$ is calculated using (3.5).

3. $\mathbf{x}_3$ is now extrapolated using $\mathbf{x}_{C+\Delta t}$ and $\mathbf{x}_C$

$$\mathbf{x}_3 = \mathbf{x}_C + \frac{(t_3 - t_C)}{\Delta t}\left(\mathbf{x}_{C+\Delta t} - \mathbf{x}_C\right) \tag{3.12}$$

Thus, in this algorithm we have 1 interpolation, 1 full state calculation and 1 extrapolation with a fixed Δt.

Algorithm VII

After Step 1 in Algorithm VI, this algorithm uses the following steps:

2. The step-size of the simulation is changed from Δt to $\{\Delta t + (t_2 - t_C)\}$ and $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]_{\{\Delta t+(t_2-t_C)\}}$ is calculated.

3. State $\mathbf{x}_3$ is calculated using (3.5).

Therefore, we have 1 interpolation, 1 full state calculation and 1 $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$ recalculation.

If the simulation time-step is changed internally as in algorithms III, V and VII, the re-calculation of $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$ may become a significant computational bottleneck if the network is large. An efficient way to circumvent on-line calculation of $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$ for a large network is to store pre-computed values for several sub-multiples of Δt ($\delta t = \Delta t/n$, n integer), in a look-up table and apply them when required. For example, if n=10, $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$ can be computed off-line 10 times for 10 values of time $(\Delta t + k\delta t)$ $k = 1, 2, \ldots 10$. Such an approach may not be precise as to the location of a switching event, however, it is sufficiently accurate compared with $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$ calculation based on a single large time-step Δt. This pre-calculation of $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$ is feasible if the network condition prior to the switching events is known. During a transient, this condition is generally unpredictable. Another approach to accommodate a large network could be to interface the power electronic circuit simulation using algorithms III, V or VII, with a fixed step-size network simulation, as reported in [?].

The above algorithms are indeed capable of correcting three switchings in one time-step. In such a case, the first two switchings are corrected initially, followed by the correction of the third switching event. Further extension for accounting more than three events is straight forward. The cost for having more switching events in one time-step is an increase in complexity of the algorithms and they become extremely cumbersome. An efficient trade-off in handling multiple switching events would be to reduce the time-step Δt such that at most two or three events occur within one Δt.

3.3.2 Variable Time-Grid Algorithms

Algorithm VIII

This algorithm is event-driven and varies its external time-grid whenever it is interrupted by a switching event. However, the internal time-step of the simulation is fixed. Whenever a switching event is encountered, the simulator is stopped at that instant and the time difference (Δh) between the initial point of that particular time step and the switching instant is calculated. Based on the value of Δh one of the following four cases is executed.

- **Case I** If $\Delta h < \Delta t/4$, an extrapolation using the two previously known states is performed to find the state at the point of switching.
- **Case II** If $\Delta h \geq \Delta t/4$ but $\leq \Delta t/2$, Backward Euler's integration is adopted for $\Delta t/4$ time-step and then an extrapolation is performed to the point of switching.
- **Case III** If $\Delta h \geq \Delta t/2$ but $\leq 3\Delta t/4$, Trapezoidal integration is adopted for $\Delta t/2$ time-step and then an extrapolation is performed to the point of switching.
- **Case IV** If $\Delta h \geq 3\Delta t/4$, Trapezoidal integration is adopted for $\Delta t/2$ time-step, Backward Euler's integration is adopted for $\Delta t/4$ time-step and then an extrapolation is performed to the point of switching.

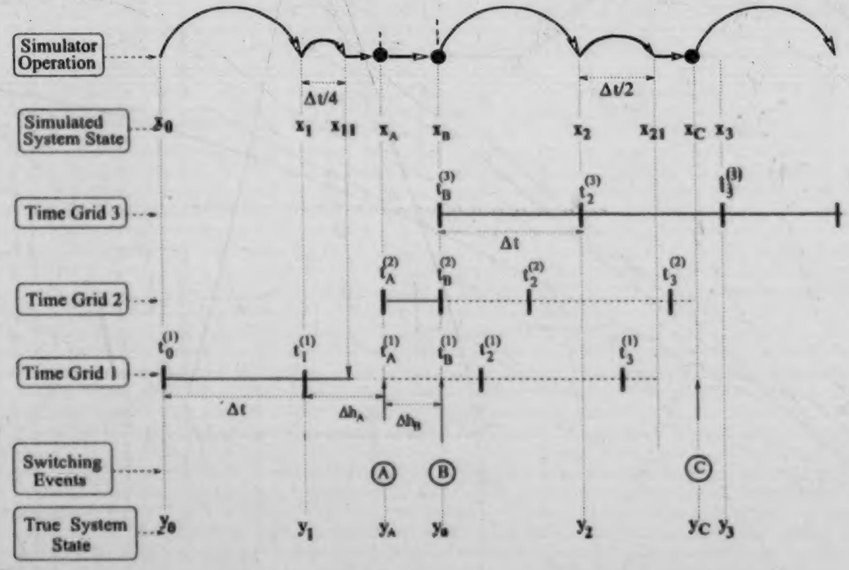


Figure 3.4: Variable time-grid Algorithm VIII for a two switching events. ⌒ : Full-state calculation, ● : Switching update, ⟶ : Linear extrapolation.

Once the simulation reaches to the switching point then, the switching change is updated and the simulation proceeds with original time-step but moves to a new time-grid. In case of two switching events, similar procedure is adopted and the simulation follows any of the four ways to reach the switching instant. For each switching event the external time-grid is changed even though the internal simulation time step remains unchanged. The only additional requirement for this algorithm is to save $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$ for a time-step of $\Delta t/2$.

Fig. 3.4 illustrates this algorithm for both single and two switching events. The simulation starts at the time $t_0^{(1)}$ on Time-Grid 1. State $\mathbf{x}_1$ at time $t_1^{(1)}$ is computed from state $\mathbf{x}_0$ with a fixed time-step Δt using (3.5). At time $t_1^{(1)}$ the simulator begins calculating the state for time $t_2^{(1)}$, however, it is interrupted by a switching event Ⓐ at time $t_A^{(1)}$. The time difference between $t_1^{(1)}$ and $t_A^{(1)}$ is less than $\Delta t/2$ but greater than $\Delta t/4$. Therefore, Case II is followed where Backward Euler yields $\mathbf{x}_{11}$ at a step of $\Delta t/4$. Then, the system state $\mathbf{x}_A$ at time $t_A^{(1)}$ is extrapolated from the state $\mathbf{x}_1$ and $\mathbf{x}_{11}$. The state at t_A is then updated with the switching events. Now, with the updated state $\mathbf{x}_A$, the simulation is transferred to Time-Grid 2 and proceeds with a fixed time-step Δt from time $t_A^{(2)}$ onward and expected to reach at time $t_2^{(2)}$. However, the simulation is interrupted again by a second switching event Ⓑ at time $t_B^{(2)}$. As the distance between $t_A^{(2)}$ and $t_B^{(2)}$ is less than $\Delta t/2$ (Case I), a single extrapolation is performed based on the state at $\mathbf{x_A}$ at $t_A^{(2)}$ and $\mathbf{x_1}$ at $t_1^{(1)}$ to find the state $\mathbf{x_B}$ at $t_B^{(2)}$. The state $\mathbf{x_B}$ at $t_B^{(2)}$ is then updated with the switching changes and the simulation moves to Time-Grid 3 and starts from $t_B^{(3)}$. From $t_B^{(3)}$ we assume that there is no switching event in the next time step and a full time-step calculation is performed to reach the state $\mathbf{x_2}$ at time $t_2^{(3)}$. From the time $t_2^{(3)}$ the simulation starts for a full step Δt but is interrupted by an event Ⓒ at time $t_C^{(3)}$. Since the distance between the time $t_2^{(3)}$ and $t_C^{(3)}$ is greater than $\Delta t/2$ but less than $3\Delta t/4$ (Case III), first a trapezoidal integration is performed to calculate the state $\mathbf{x_{21}}$ at $t_2^{(3)}+\Delta t/2$. Thereafter, linear extrapolation between $\mathbf{x_2}$ and $\mathbf{x_{21}}$ is performed to reach the state $\mathbf{x_C}$ at $t_C^{(3)}$. The simulator then updates the switching changes and moves to a new time-grid (not shown in the figure). The simulation follows the similar approach for all the future switching events and the states are recorded. Clearly, the output data of the simulation will be unevenly spaced due to the changes in the original time-grid. Linear Interpolation is used to get an evenly spaced data set.

The benefits of implementing this variable time grid algorithm are as follows:

- No Post Event Correction is necessary as the algorithm always uses the corrected states.
- The algorithm is not suitable for implementing in real-time as it needs the information about switching events at any time-step even before the calculation of that time-step begins.
- The internal simulation time step remains always unchanged and re-calculation of $[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$ is obviated.

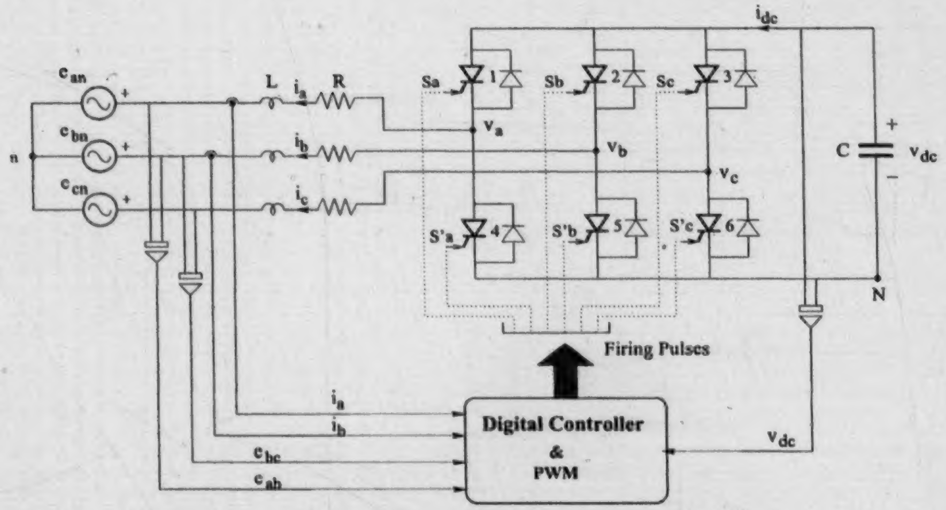


Figure 3.5: PWM VSC system and its digital controller

3.4 Case Study: PWM VSC System

3.4.1 System Model

Fig. 3.5 illustrates a typical set-up of the PWM VSC or alternately known as STATCOM system and its digital controller. The system consists of a three-phase IGBT bridge (switching frequency $f_{sw} = 1\text{kHz}$) connected via a series impedance ($R = 0.5\Omega$ and $L = 3.0\text{mH}$) to the AC bus assumed to be a balanced voltage source ($E = 110\text{V rms } ll$, $f = 60\text{Hz}$). L represents the filter inductance and the leakage inductance of the converter transformer. R accounts for the converter and transformer conduction losses. The dc side capacitor is $C = 4900\mu\text{F}$. The switches in the VSC are modeled as ideal bi-directional switches with gate turn-on and turn-off controls. The VSC model is based on discrete switching functions. Three switching signals S_k, $k = \{a, b, c\}$ control the upper switches in each leg of the converter. The lower switches in each leg of the converter are switched in a complementary manner. Accordingly, the output voltages of the converter with respect to the negative DC bus N are given as

$$v_{kN} = S_k \, v_{dc} \; ; \; k = \{a, b, c\} \tag{3.13}$$

Under balanced conditions the converter output voltages with respect to the AC system neutral n are given as

$$v_{kn} = \frac{2}{3} v_{kN} - \frac{1}{3} \sum_{\substack{i=\{a,b,c\} \\ i \neq k}} v_{iN} \tag{3.14}$$

Taking i_a, i_b and v_{dc} as the states, the time-domain model of the system can be represented by three differential equations

$$\frac{di_a}{dt} = -(R/L)\, i_a + (1/L)\, (v_{an} - e_{an}) \tag{3.15}$$

$$\frac{di_b}{dt} = -(R/L)\, i_b + (1/L)\, (v_{bn} - e_{bn}) \tag{3.16}$$

$$\frac{dv_{dc}}{dt} = -(1/C)\, (i_{dc}) \tag{3.17}$$

The current in phase c, can be expressed as a linear combination of i_a and i_b. The DC side current is given as

$$i_{dc} = \sum_{k=\{a,b,c\}} S_k\, i_k \tag{3.18}$$

The above equations are solved by the simulator at every time-step Δt.

3.4.2 Digital Controller

The control design [50] has been carried out in the synchronous dq frame. The control algorithm is executed every T_s which is the controller sampling period. First a coordinate transformation from abc frame to dq frame is performed on the system signals. Then a decoupled control of the currents i_d and i_q is performed using P-I compensator's for each of the current loops. The DC-link voltage is regulated through an external feedback loop. The resulting control quantities- modulation index and phase angle of the control signal are then sent to the Pulse Width Modulator (PWM) for the generation of switching signals based on the sampling technique approach [51]. A delay of one sampling period T_s is introduced in the controller implementation in order to account for the finite execution time that would be taken by the hardware. The detail design procedure of the controller would be described in the next chapter when a complete design of a digital controller for a VSC HVDC system will be carried out.

3.5 Results and Discussion

All the algorithms, I through VIII have been used to simulate the PWM VSC system and their performance was compared with that of a fixed time-step algorithm (Fig. 3.1). Results were obtained through off-line simulation of programs written in C language. It was observed that as the computational complexity of the algorithms decreased, their execution speed increased, however, at the cost of a small inaccuracy. Both open-loop and closed-loop control studies were performed. In the modulator, sinusoidal PWM scheme with 1kHz switching frequency was used in which only half of the carrier signal was implemented in one sampling period T_s of the digital controller i.e., $T_s = (1/2)T_{sw}$ or, $f_s = 2f_{sw}$ where f_{sw} =1 kHz. Such an approach was taken in order to limit the number of discrete switching events to a maximum of three in one simulation time-step Δt.

Fig. 3.6 illustrates the multiple switching phenomenon using one period of the triangular carrier used in PWM. The snapshot of PWM produced here starts at time t=2500 μs and ends at t=3500 μs with Δt= 100 μs. The control signals computed for the three phases intersect with the first leg of the carrier signal at t=2550 μs (phase c), t=2829 μs (phase b) and t=2869 μs (phase a). The switching signals S_k (k=a, b, c) are also shown in Fig. 3.6. Evidently, the switching of phase c is the only event during the time interval t=2500 μs to t=2600 μs, whereas switching of phase a and phase b constitute two events in the time interval t=2800 μs to t=2900 μs. Similar behavior can be seen during the second leg of the carrier signal. It can also be observed that the case of two switching events occurs when two control values are close in their magnitude. As the time-step Δt is increased, multiple switchings occur more often. This fact is corroborated in Table I which records the frequency of time-steps with no events, one event and two events.

3.5.1 Open Loop Control

Under open-loop conditions the output of the VSC was varied by varying the modulation index $m_a(= V_{control}/V_{tri})$ and the phase angle δ of the control signal. The value m_a = 0.8 and δ = 10^o were chosen in order to get high fundamental

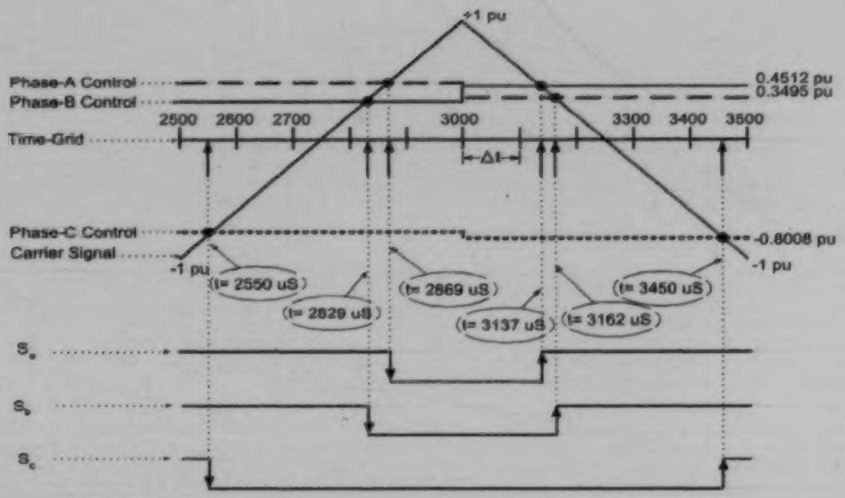


Figure 3.6: Illustration of multiple switchings through a detailed view of the simulation for one complete cycle of control signal suing D-STATCOM test system.

error and THD using fixed time-step algorithm so that the improvements due to the corrected algorithms can be clearly demonstrated. To calculate the fundamental error, the reference is chosen as a case of time-step, $\Delta t = 1\mu s$. This Δt is 1000 times smaller than the switching time period $T_s(= 500\mu s\ )$ and it has been selected to serve as a benchmark against which the results obtained with limited step-sizes $\Delta t = (10, 50, 100, 150\mu s)$ were compared. Results for the fixed time-step approach were also verified against simulations on PSCAD/EMTDC Version 3.2.

The two major effects observed due to the changes in Δt were:

1. A significant change in the fundamental component of the phase current.

Table 3.1: Variation of switching events with simulation step-size for a total simulation period of 2 s with carrier frequency of 1 kHz.

Δt μs	Time-steps with 0 events		Time-steps with 1 event		Time-steps with 2 events	
	No.	%	No.	%	No.	%
10	188038	94.02	11926	5.96	34	0.02
50	28054	70.14	11895	29.74	49	0.12
100	10086	50.43	7833	39.17	2080	10.40
200	1767	17.67	5924	59.25	2308	23.08

2. A marked crowding of the frequency spectrum with non-characteristic harmonics and hence a degradation in THD.

The change in fundamental component is expressed in terms of percentage absolute error which is defined as;

$$\varepsilon = \left(\frac{|(I_1)_{\Delta t} - (I_1)_{1\mu s}|}{(I_1)_{1\mu s}} \right) 100 \tag{3.19}$$

where, $(I_1)_{\Delta t}$ is the fundamental component of current at a particular time-step and $(I_1)_{1\mu s}$ is the fundamental component of current with 1 μs time-step. The percentage THD for the current waveform simulated using a certain time-step is defined as:

$$\%THD = \left(\sqrt{\sum_{h \neq 1} \left(\frac{I_h}{I_1} \right)^2} \right) 100 \tag{3.20}$$

For the fixed time-step algorithm $(I_1)_{1\mu s} = 17.28$ A and the THD = 16.28 %. The ε and THD for different algorithms with step-sizes of 10, 50, 100, and 150 μs are also presented in Table 3.2. The table also summarizes the number of operations required by each of the proposed algorithms. Algorithm I and II are the most rigorous and accurate while other algorithms bear relatively less computational burden and are therefore less accurate in comparison. The accuracy of these algorithms (III through VII) can be further improved through a correction of $\mathbf{x_2}$ state by interpolation. A remarkable improvement in performance especially in THD was observed when state $\mathbf{x_2}$ was corrected (as seen from Table 3.2).

The following conclusions can be drawn from the results:

- In the fixed time-step algorithm, as the time-step is increased from $\Delta t = 10\mu s$ to $\Delta t = 100\mu s$, we find that the fundamental current increases drastically (from 17.1 A to 24.7 A), due to not accounting the switching events at their right instants, hence the percentage fundamental error ε increases from 2.4% to 33.66%. The harmonic current also increases at almost the same rate. Therefore, THD remains almost unchanged (around 16%). However, at much higher time-steps e.g., $\Delta t = 150\mu s$, the harmonic current increases at a faster rate than the fundamental current, which is why we get a higher THD (46%) at that time-step.

Table 3.2: Fundamental error ε and THD of I_a for varying Δt.

Algorithms	$\Delta t = 10\mu s$		$\Delta t = 50\mu s$		$\Delta t = 100\mu s$		$\Delta t = 150\mu s$	
	ε (%)	THD(%)	ε (%)	THD(%)	ε (%)	THD(%)	ε (%)	THD(%)
Fixed Δt	2.40	16.19	16.95	16.85	33.66	16.64	38.25	46.15
I	0.30	16.33	0.58	16.36	0.92	16.32	3.10	19.22
II	0.23	16.42	0.40	18.72	3.80	29.03	0.57	35.45
III	0.29	16.43	0.35	18.78	9.30	28.75	4.80	36.32
IV	0.29	16.43	0.34	18.72	7.60	37.32	1.10	38.03
V	0.29	16.43	0.34	18.78	6.25	36.47	6.01	39.73
VI	0.24	16.37	0.98	18.63	4.80	29.03	0.75	37.46
VII	0.17	16.37	0.63	18.64	3.72	28.58	3.61	38.00
VIII	0.80	16.29	1.30	17.93	1.01	27.5	6.8	40.51

- A time-step of over $50\mu s$ (for the study system with a switching frequency of 1 kHz) is inadequate for simulation using the fixed time-step approach due to a high error (the error increases to 38.25% for $\Delta t = 150\mu s$.

- As the time-step varies from $\Delta t = 10\mu s$ to $\Delta t = 100\mu s$, Algorithm I treats the switching events accurately, thereby reducing both the fundamental and the harmonic currents. Therefore, both the percentage rms error and THD remain small (close to their values at $\Delta t = 10\mu s$.

- For Algorithms II through VII, both THD and ε increase, with the increase of time-step, however, the increase in ε is less pronounced compared to that for the fixed time-step algorithm. Even at a smaller time-step such as $\Delta t = 10\mu s$, the error is less than 0.8% for all the proposed algorithms, whereas fixed time-step algorithm shows an error of 2.4%. The fundamental error ε decreases to a range of 0.5% - 9% from 38.2% for the fixed time-step algorithm when $\Delta t = 150\mu s$.

- Algorithms II through VII are less accurate in comparison with Algorithm I, and therefore produce significant harmonic current at higher time-steps. However, they are still able to compensate for the fundamental current and keep its value low. Therefore, the percentage rms error is small but the THD for these algorithms remains high at higher time-steps. Furthermore, Algorithms II to VII do not use the corrected state $\mathbf{x_2}$ as the the output at time t_2 in order to

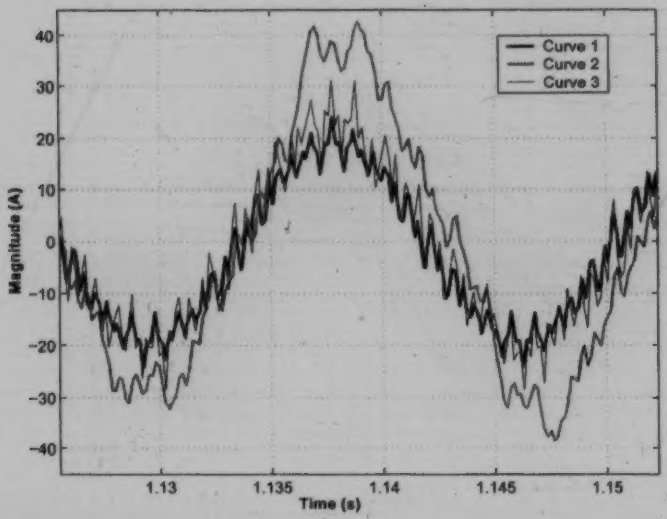


Figure 3.7: Detailed view of the steady state current I_a. Curve 1: Fixed time-step algorithm with $\Delta t = 10\mu s$,Curve 2: Fixed time-step algorithm with $\Delta t = 150\mu s$, Curve 3: Algorithm VII with $\Delta t = 150\mu s$.

conform to the real-time condition that it would not be possible to send the corrected value to the controller.

- If state $\mathbf{x_2}$ is allowed to be corrected, a significant improvement in THD is observed. The results in Table II for a 150 μs time-step reveal that all algorithms (except Algorithm VIII) show THD in the range of 18% to 24%, whereas fixed time-step algorithm shows a THD of 46%. Off-line simulators can exploit the benefit of correcting state $\mathbf{x_2}$.

For any given Δt, the accuracy of all the algorithms was found to be similar to that of the fixed time-step algorithm using one-tenth Δt. Using similar simulation parameters and environment, the algorithm used in HYPERSIM [45] has been extended for multiple switching events. A comparative study revealed that all the proposed algorithms produce results that are in close agreement with the results obtained using the HYPERSIM algorithm until Δt = 100 μs. At 10, 50, and 100 μs time-steps the HYPERSIM algorithm yielded an error of 0.34, 0.98 and 4.6 (%) respectively, however, at 150 μs or higher it produced an increasing error (higher than

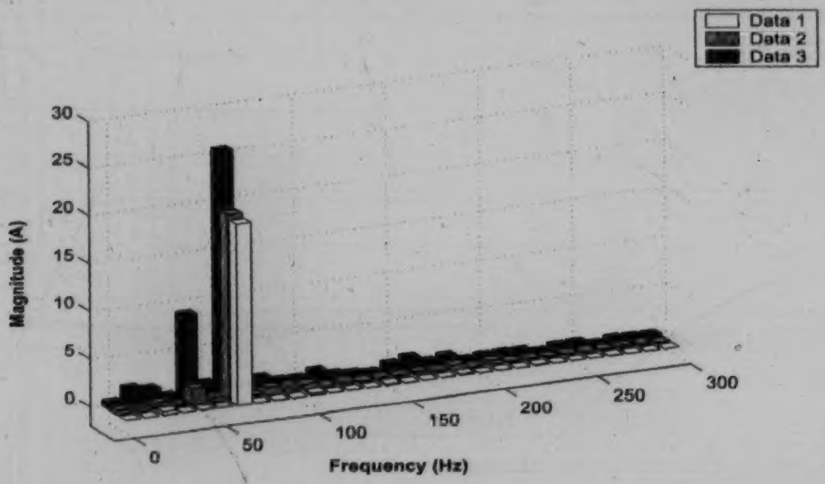


Figure 3.8: Frequency spectrum of I_a. Data 1: Fixed time-step algorithm with $\triangle t = 10\mu s$, Data 2: Algorithm VII with $\triangle t = 150\mu s$, Data 3; Fixed time-step algorithm with $\triangle t = 150\mu s$.

30%).

A comparison of steady-state current for fixed time-step algorithm and Algorithm VII is shown in Fig. 3.7. The current waveform with fixed time-step algorithm differs significantly when the time-step is changed from $10\mu s$ to $150\mu s$. However Algorithm VII with $150\mu s$ is in close conformity with the $10\mu s$ case of fixed time-step algorithm. The difference is more obvious from the Fig. 3.8, where frequency spectrum is plotted. Fixed time-step algorithm with $\triangle t = 150\mu s$ shows a substantial difference in fundamental component in comparison with the other two cases. It also induces non-characteristic harmonics which eventually result in higher THD.

The behavior of various algorithms with the increase in $\triangle t$ can be further understood from Fig. 3.9 where the ε has been plotted with respect to $\triangle t$. It is clear that for any value of time-step, all the algorithms produce significantly less error than that of fixed time-step. Some algorithms show a little increase in error at a time-step of $100\mu s$. Algorithm I and III produces the best result for a time-step of $150\mu s$ and as expected the error becomes high for other algorithms, however, it is much less than that of fixed time-step algorithm. Therefore, it can be concluded that all the algorithms substantially improve the simulation accuracy.

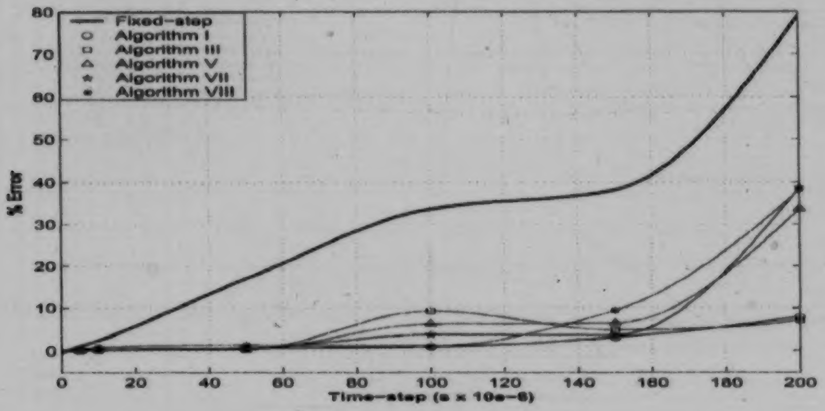


Figure 3.9: Plot of percentage fundamental error ε with respect to simulation time-step

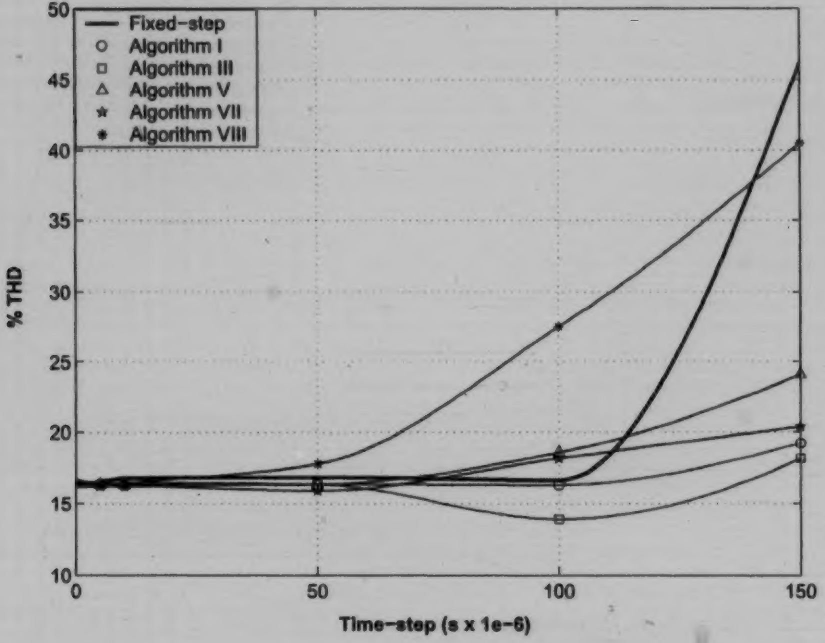


Figure 3.10: Plot of THD with respect to simulation time-step

The variation of THD as a function of time-step for various algorithms using corrected state $\mathbf{x}_2$ is shown in Fig. 3.10. Fixed time-step algorithm shows almost same THD until $\Delta t = 100\mu s$ after which it increases dramatically. On the other

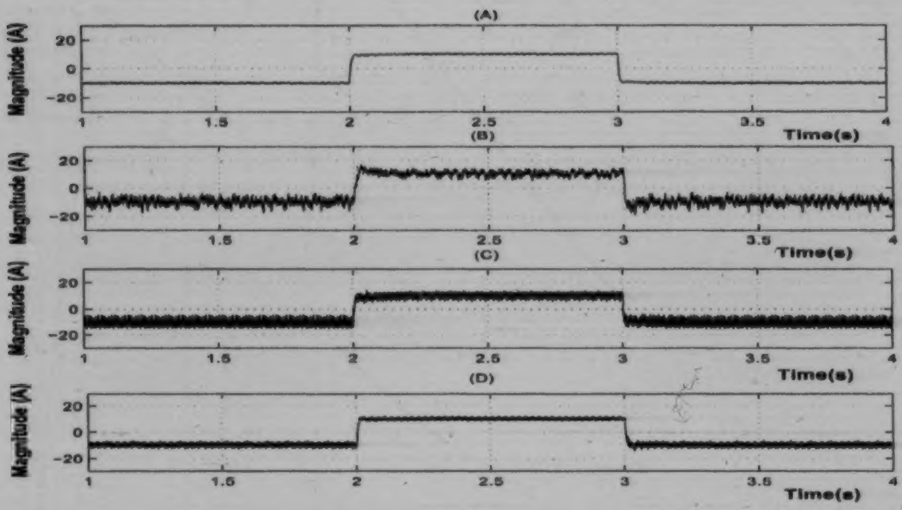


Figure 3.11: Step response in i_q simulated using the (A) Fixed time-step algorithm with $\triangle t = 10\mu s$ (B) Fixed time-step algorithm with $\triangle t = 100\mu s$, (C) Algorithm VII with $\triangle t = 100\mu s$ and (D) Algorithm VII with $\triangle t = 100\mu s$ but $\mathbf{x_2}$ corrected.

hand, most of the algorithms (except Algorithm VIII) show a lower THD than the case of fixed time-step algorithm. Algorithm VIII produces a higher ε and THD compared to the fixed time-grid algorithms for a large time-step of $150\mu s$ or higher. The reason is that the algorithm resorts to frequent application of Backward Euler and linear extrapolation (due to an increased step-size resulting in more switching events within a time-step) which is less accurate than the trapezoidal technique. However, it produces results as good as or even better than other algorithms at smaller time-steps ($< 100\ \mu s$).

3.5.2 Closed-Loop Control

Fig. 3.11 shows the closed loop response of the PWM VSC. A step response in i_q was simulated. The simulations were carried out for all the algorithms using various step-sizes, however, results are shown for the fixed step-size algorithm for $\triangle t = 10\mu s$ and for $\triangle t = 100\mu s$ and Algorithm VII for $\triangle t = 100\mu s$. The predominant effect observed due to an increase in the time step was the appearance of noise in the control quantities. As seen from Fig. 3.11, the traces of i_q for $\triangle t = 10\mu s$ are smooth

Table 3.3: Comparison of MAD and SNR (dB) of i_q produced by different algorithms for $\triangle$t=100 μs

Algorithms	x_2 Uncorrected		x_2 Corrected	
	MAD	**SNR**	**MAD**	**SNR**
Fixed $\triangle$t	1.79	29.32	-	-
I	-	-	0.05	101.4
II	1.32	34.8	0.06	96.32
III	1.37	35.32	0.52	55.24
IV	1.62	32.78	0.63	50.14
V	1.6	32.06	0.76	49.36
VI	1.61	33.54	0.65	49.38
VII	1.6	33.54	0.64	49.56
VIII	1.8	30.1	-	-

whereas those for $\Delta t = 100\mu s$ contain a noise superimposed on the average value. A Fourier analysis proved that the noise is mainly white with unlimited bandwidth. Table 3.3 compares the Mean Absolute Deviation (MAD) and the Signal-to-Noise Ratio (SNR $= 10\log_{10}\left(\frac{P_S}{P_N}\right)$ dB, where P_S is the signal power and P_N is the noise power) of i_q for all the algorithms. Algorithm I shows the best signal quality as it always uses the true state results. Results in Table 3.3 reveal that the signal quality deteriorates slightly as we move from Algorithm II to Algorithm VII. It can be seen that (Fig. 3.11(C)) for Algorithm VII using $\Delta t = 100\mu s$, the output i_q shows a MAD of 1.6 around its reference value (10 A), however, the fixed time-step algorithm shows a MAD of 1.79. The SNR for Algorithm VII is 33.54 dB and it is 29.32 dB for the fixed time-step algorithm. Table 3.3 reveals that if corrected states are used for control, MAD would be reduced to 0.64 and SNR would be increased to 49.56 dB. In a practical set-up of the digital controller the noise may be reduced by using low-pass anti-aliasing filters on the system signals before they are sampled. In an off-line simulation program the noise in the control quantities may also be substantially reduced by using linear interpolation in the control algorithm.

3.5.3 Execution Time

The execution time required for the off-line simulation of these algorithms depends on several factors such as the processor speed, hardware (such as RAM and bus speed), and the operating system. Notwithstanding the minor variations in processor time due to multi-tasking, the testing conditions for all the algorithms were maintained the same. The simulation programs were coded in C and all the programs were executed on the same hardware using a 1.5 GHz Pentium IV processor running Windows 2000. The CPU time required to simulate the test system for the worst case of multiple switching (two switching events in one time-step) is shown in Fig 3.12. The increased complexity in correcting the switching events using different algorithms increases the execution time by a maximum of only 9% (Algorithm I), compared to the fixed time-step approach. However, the improvement in the simulation accuracy is significant.

Using the proposed algorithms, an accurate simulation is possible even with a larger step-size, which makes these algorithms suitable for real-time application. Even though the difference in CPU execution time among different algorithms is not very large, it has been found that algorithms that do not involve rigorous calculations (II through VII) as compared to Algorithm I, requires less CPU time; for example, the CPU time requirement for Algorithm VII is 7.84% lower than that of Algorithm I. Algorithm VIII takes the least amount of time of all the proposed algorithms. The D-STATCOM system, due to its modeling simplicity is not able to show a larger CPU time difference between algorithms, however, a larger system with more modeling complexity such as multi-pulse HVDC system would be able to show definite computational advantages in favour of Algorithm II through VII. And there lies the motivation of using the proposed algorithms even with uncorrected states.

3.6 Summary

This chapter discussed the issues related to inter-step switching events especially multiple switching events and provides algorithms suitable for the accounting of such events accurately and efficiently in digital simulation of power electronic systems. A Pulse Width Modulated (PWM) Voltage Source Converter (VSC) based STATCOM system is used for simulation studies. Results obtained using these algorithms indicate excellent performance in comparison with the fixed time-step algorithm. The following points can be concluded:

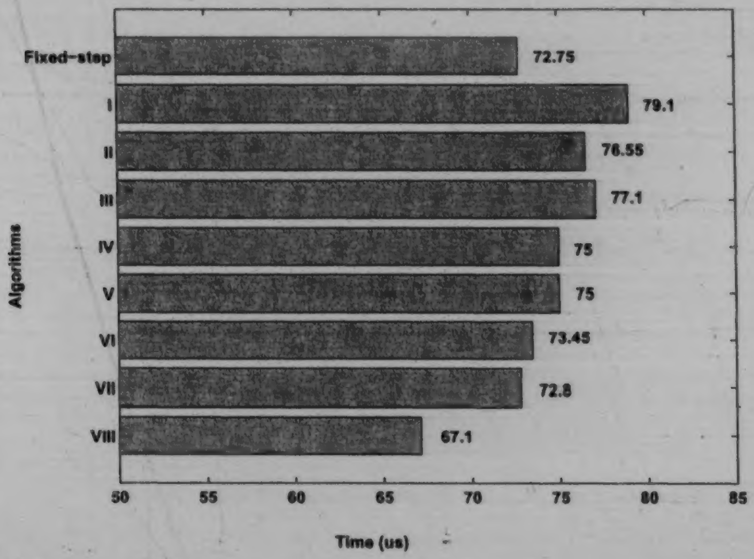


Figure 3.12: Execution time for the time-step involving multiple switching events using various algorithms with $\Delta t = 100\ \mu s$

- Algorithms clearly describe how multiple switching events are handled in digital simulation. A full comparison of the performance of the proposed algorithms has been presented. All algorithms show significant improvement in accuracy under both open-loop and closed-loop control.

- Despite having real-time constraints such as use of uncorrected states in the control, the performance of the proposed algorithms was found to be very good.

- It is important to note that the improvement in accuracy of the algorithms comes at a cost of only a 9% (maximum) increase in execution time in comparison with the fixed time-step algorithm. The algorithms presented here are amenable for real-time implementation and are implemented in HIL simulation of VSC-HVDC using the PC-cluster in our RTX-LAB (**R**eal-**T**ime e**X**perimental Lab). Of these algorithms, Algorithm II is chosen for both off-line and HIL simulation of VSC-HVDC system due to its high accuracy with comparatively low execution time.

Chapter 4

Modeling, Control and Off-line Simulation of VSC-HVDC

4.1 Introduction

The design, performance evaluation, and interactive stability studies of systems like VSC-HVDC need extensive analysis using reliable and accurate digital simulators. PSCAD/EMTDC and MATLAB/SIMULINK are two such off-line digital time-domain simulators used now-a-days for studying different AC and DC power systems. Both programs can deal with power systems containing power electronics apparatus with accuracy and reliability, therefore, found to be used increasingly in both academia and power industry. Both programs allow the user to construct schematic diagrams of electrical networks, run the simulation, and produce the results in a graphical environment. This chapter[1] focuses on:

- Modeling and simulation of VSC-HVDC system using two simulation tools PSCAD/EMTDC and MATLAB/SIMULINK.
- Design of a digital controller for the VSC-HVDC system using a discretized plant model.
- Comparative study of simulation results for both transient and steady-state situations.

---

[1] Some material of this chapter has been published: M. O. Faruque, Y. Zhang, V. Dinavahi, "Detailed Modeling of CIGRE HVDC Benchmark System Using PSCAD/EMTDC and PSB/SIMULINK", *IEEE Trans. on Power Delivery*, Vol. 21, No. 1, Jan. 2006, pp. 378-387.

The modeling of the systems and their controllers and the use of different discretization methods, directly affects the accuracy and consistency of the simulation results. Therefore, detailed attention is a prerequisite when modeling the system, converter bridges and controls. Besides selecting the accurate model for the system, selection of suitable solution algorithm is also important. A comparative study using various simulation softwares will not only assist to pinpoint the pros and cons of the programs but will also help understand the behavior and interaction of the system being simulated, especially, when it goes through various contingency situations. Moreover, it helps in designing the control algorithms through offering a test bed for running the control scheme that later can be used for real-time hardware-in-the-loop simulation. Therefore, an off-line simulation of VSC-HVDC and a comparison between the PSCAD/EMTDC, MATLAB/SIMULINK have been carried out in this chapter. Care has been taken to ensure the use of similar modeling and control techniques and same system parameters for the study.

4.2 VSC-HVDC Test System

VSC-HVDC is a recently proposed DC transmission technique which is based on voltage-source converter technology where force-commutated or self-commutated switching devices such as GTOs or IGBTs are operated through Pulse Width Modulation (PWM) scheme. With this technique, it is possible to generate voltages at any magnitude, angle and frequency (up to a certain limit) through changes in the control and PWM scheme. Because of the controllability of the voltage source converters, active power can be transferred from one side to another and reactive power can be generated independently. With the object of applying the developed modeling and solution techniques, the switching event synchronization technique and the control scheme in the simulation, a fairly complex system such as the VSC-HVDC system has been chosen as a test system to carry out the research.

Fig. 4.1 shows the schematic diagram of the VSC-HVDC test system which mainly consists of two AC sources, phase reactors on both sides, two converters and a DC-link. The two AC supplies of the network is represented by three-phase Y connected

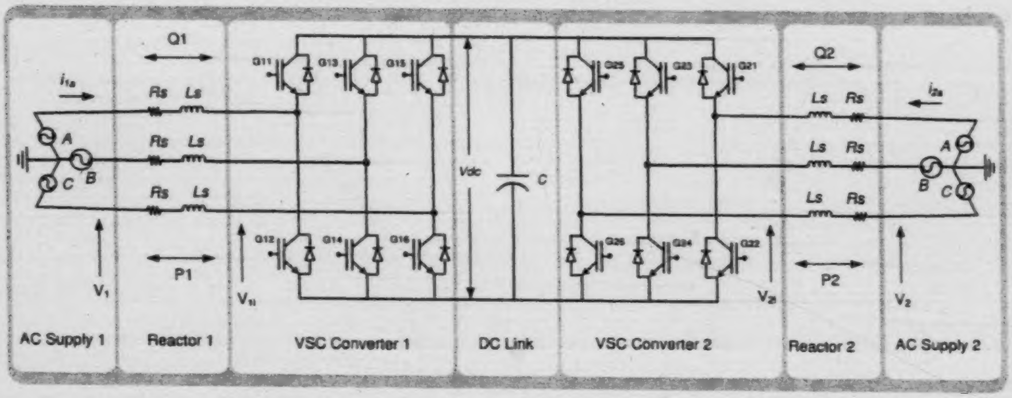


Figure 4.1: Schematic diagram of VSC-HVDC test system.

and grounded infinite sources with no source impedance. The two converters, one operating as an inverter and the other as rectifier, are built of IGBTs and connected back-to-back through a DC-link capacitor. When active power flow is reversed, the converters interchange their function, i.e. the one working as a rectifier becomes an inverter and vice versa. The converters are three-phase, two-level and six-pulse bridges. The DC capacitor serves as energy storage for controlling the power flow and also reduces the ripple on the DC side. The phase reactors are used for controlling both active and reactive power flow through regulating the current flow. It also serves as an AC filter to reduce the high frequency harmonic contents of the AC currents caused by the high frequency switchings.

4.2.1 Operation of VSC-HVDC

The converters used in a VSC-HVDC system can be considered as variable voltage sources as they are capable of generating voltages at any magnitude, phase-angle and frequency. With respect to a DC-link voltage V_{dc}, the instantaneous voltage at the terminal of the converter 2, V_{2i} can be given as:

$$V_{2i} = K m_a V_{dc} \sin(\omega t + \delta) \tag{4.1}$$

where, K is the constant whose value depends on the modulation scheme, m_a is the modulation index defined by the ratio of the peak value of the modulating wave to the

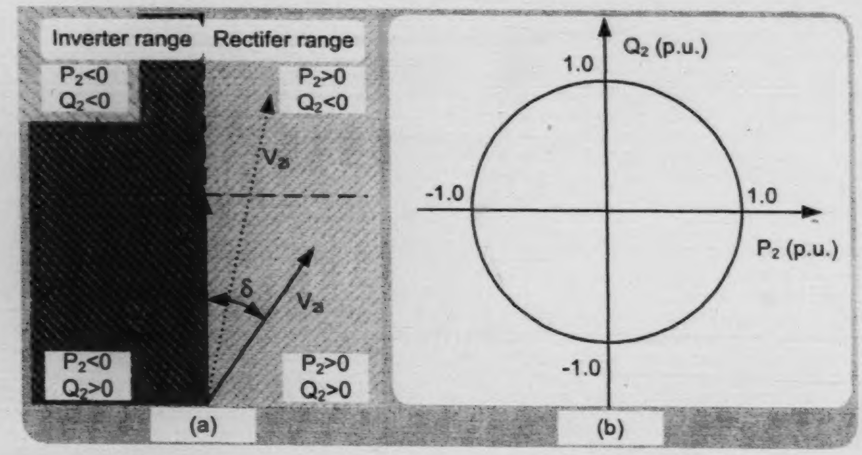


Figure 4.2: (a) Phasor diagram indicating the changes in power flow due to the changes in the magnitude and angle of $V2i$, (b) Locus of maximum P_2 and Q_2 obtained through the variation of $V2i$.

peak value of the carrier wave; ω is the fundamental frequency and δ is the phase shift with respect to the AC terminal voltage. The phase-shift angle, δ and the modulation index, m_a can be controlled to generate any combination of voltage magnitude and phase angle (limited by the DC voltage). Thus, the active and reactive power flow between the converter and the network through a lossless reactor, X, can be calculated using the power equations:

$$P_2 = \frac{V_{2i} V_2 \sin\delta}{X} \tag{4.2}$$

$$Q_2 = \frac{V_{2i}(V_2 \cos\delta - V_{2i})}{X} \tag{4.3}$$

Fig. 4.2(a) shows how power flow can be controlled by varying the amplitude and phase angle of the converter voltage, V_{2i}. Considering the AC line voltage, V_2 as constant and if the notation of power flow to the converter is assumed positive, for any leading angle of δ, the active power, P_2 is negative, i.e., active power will flow from the DC-link to the grid line. In such case, the converter will act as a PWM inverter. Similarly, if V_{2i} lags the AC voltage, V_2, the active power flow is considered positive and the power will flow from the grid line to the DC-link. The reactive power flow, Q_2 is controlled by the magnitude of V_{2i} in such a way that if V_{2i} is greater than V_2, Q_2 is negative which means reactive power is supplied to the network (capacitive), if not, Q_2 will be absorbed by the converter (inductive).

In a VSC-HVDC link, the active power flow on the AC side is equal to the power transmitted through the DC link, if all the losses (switching loss, resistive loss) are neglected [70]. The reactive power flow can be produced independently as required by the network. Fig. 4.2(b) shows the locus of the active and reactive power flow where the maximum power flow is considered as 1.0 p.u. based on the assumption that the two AC side at both ends are at 1.0 p.u. voltage. The VSC-HVDC is able to operate at any point within the circle while the radius of the circle will represent the MVA rating of the converter.

4.3 Control of VSC-HVDC

A VSC-HVDC is capable of controlling the active power flow, the reactive power flow, the AC voltages at both converters, the DC voltage and the frequency. The most important aspect of a VSC-HVDC is its ability to control active and reactive power independently. The active power is controlled by controlling the DC-link voltage and the reactive power is controlled by changing the AC voltages at the converters terminals. The control of VSC-HVDC converters are based on a fast inner current control loop which controls the AC currents of the system where the AC current references are supplied by a outer controller. The outer controller could be the DC voltage controller, the AC voltage controller, the reactive power controller, the active power controller or the frequency controller. In this control scheme, the outer controller for ***converter 1*** is a DC voltage controller, whereas for ***converter 2***, a linear relationship, where the reference current signals for the inner current controllers are generated from the reference values of active and reactive power. However, a robust power tracking controller can be easily implemented in the outer loop to generate the required current references.

4.3.1 Control of Converter 1

The objective of the control of converter 1 is to keep the DC-link voltage constant regardless of the magnitude and direction of power flow. As the supply voltage and

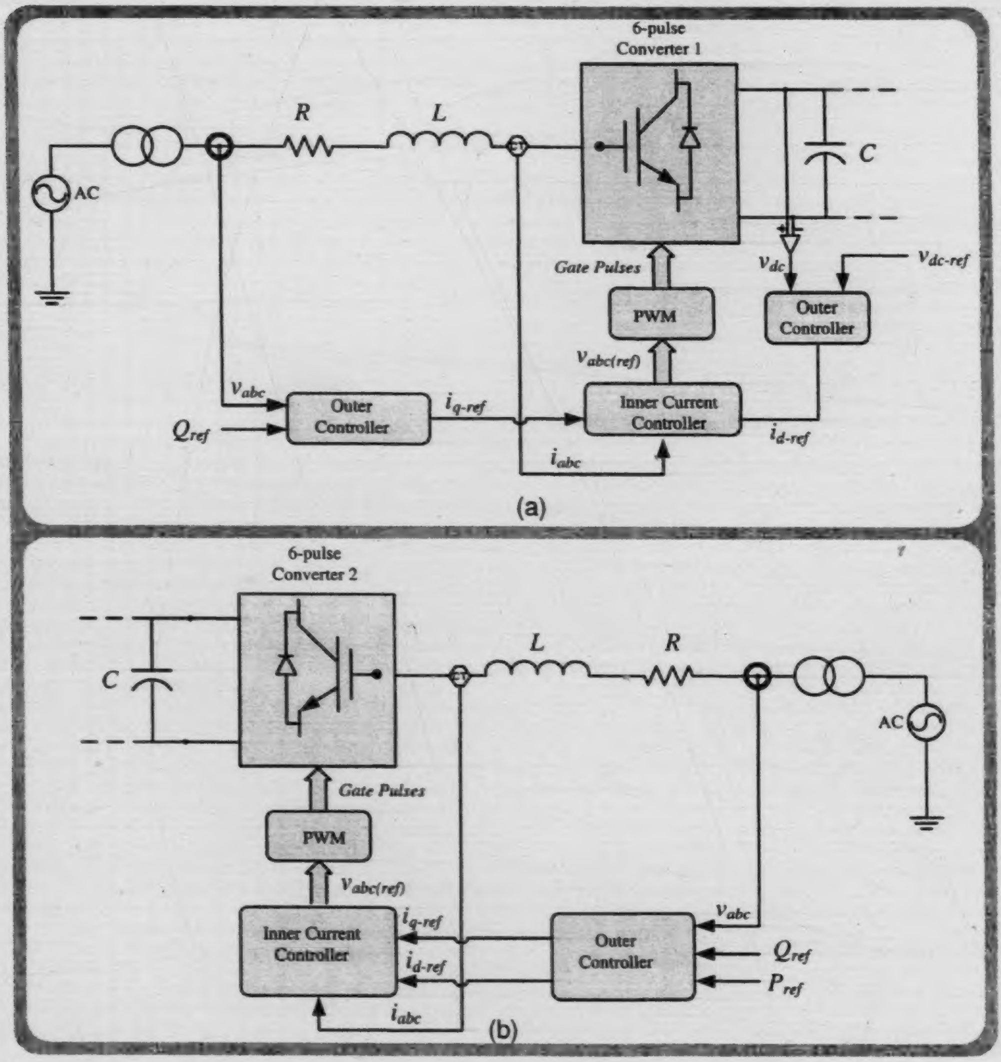


Figure 4.3: Overall control scheme of the VSC-HVDC (a) Control of converter 1 (b) Control of converter 2

the line currents are assumed to be balanced, and their sum of instantaneous values is zero, vector control or decoupled control technique can be implemented using the orthogonal co-ordinate system in which vectors will be described in the *dq* frame. In the *dq* frame, these vectors become DC value which facilitates the use of PI controllers for error tracking. Conversion of *abc* phasors to *dq* platform is done through two steps:

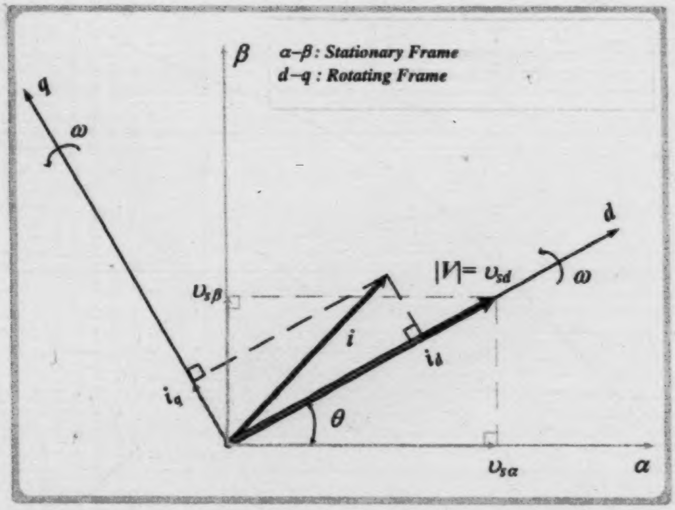


Figure 4.4: Vector diagram of coordinate transformation system.

in the first step the three phase system is transformed into a stationary orthogonal two phase $\alpha\beta$ system where the two vectors are spaced by 90°. The vector diagram showing $abc - \alpha\beta - dq$ coordinates is given in Fig. 4.4. The transformations for voltages and currents are done by using the following equations:

$$\begin{bmatrix} v_\alpha \\ v_\beta \\ v_0 \end{bmatrix} = C_1 \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}, \begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix} = C_1 \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{4.4}$$

where,

$$C_1 = \frac{2}{3} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/2 & 1/2 & 12 \end{bmatrix} \tag{4.5}$$

For a balanced system, v_0 and i_0 are equal to zero and hence,

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} 2/3 & 1/3 \\ 0 & 1/\sqrt{3} \end{bmatrix} \begin{bmatrix} v_{ab} \\ v_{bc} \end{bmatrix} \tag{4.6}$$

and

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1/\sqrt{3} & 2/3 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \end{bmatrix} \tag{4.7}$$

The second transformation, $\alpha\beta$ to dq expresses the voltages and currents from a stationary frame to a synchronously rotating frame. The d axis is always in coincidence with the synchronous voltage vector and the q axis is in quadrature with it. The d-axis current component, i_d, controls the instantaneous active power and the q-axis current component, i_q, dictates the reactive power. They follow the trajectory of voltage vector and are given by the following time-varying transformations.

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = C_2(t) \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}, \begin{bmatrix} i_d \\ i_q \end{bmatrix} = C_2(t) \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{4.8}$$

where,

$$C_2(t) = \begin{bmatrix} \cos\theta(t) & \sin\theta(t) \\ -\sin\theta(t) & \cos\theta(t) \end{bmatrix} \tag{4.9}$$

For a balanced and steady-state situation, the dq vectors will be constant and therefore, can be tracked using PI controllers.

4.3.1.1 PWM and Gate Pulse Generation

Pulse Width Modulation (PWM) is one of the most common techniques employed in generating gate pulses for the operation of the switching devices used in the converters. In this technique, generally, gate pulses are generated through the comparison of a time varying modulating signal with a triangular carrier signal as shown in Fig. 4.5. Fig. 4.5 (a) shows the comparison of discretized modulating signal $v_{mod} = V\sin(\omega t+\delta)$ with the triangular carrier signal, V_{tri}, where, in each crossing of them, the pulse flips its state from 1 to 0 or 0 to 1, and thereby, produces the gate pulses for one switching device and shown in Fig. 4.5(b). When modulating signal amplitude is greater than the carrier amplitude, $S = 1$ and when modulating signal amplitude is less than the carrier amplitude, $S = 0$. For a three phase system, three modulating signals with a phase difference of 120°, will be compared with the same triangular carrier to generate three sets of gate pulses $S_k(k = a, b, c)$ for three legs of a six-pulse converter. The gate signals for the two switches of one leg of the converter are the compliments of each other. The frequency of the triangular signal will be the converter switching frequency f_{sw} and the frequency of the modulating signal will be the fundamental frequency f_1 of the converter output. The ratio of the carrier frequency to the modulating signal frequency, $m_f = f_{sw}/f_1$ is known as the frequency modulation index and the ratio of amplitude of modulating signal to the amplitude of carrier signal is called amplitude modulation index, $m_a = V_{mod}/V_{tri}$. In the VSC-HVDC system, m_f is kept constant

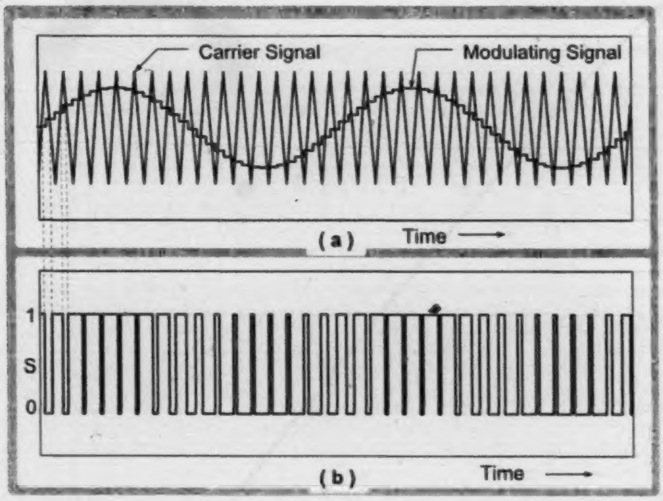


Figure 4.5: Sinusoidal PWM: (a) Comparison of modulating signal and the carrier, (b) Generated pulse for one switching device.

but m_a and the phase angle, δ changes with the requirement of power level. Keeping a constant DC voltage, V_{dc}, and using sinusoidal-PWM (SPWM), a switched pulses of DC voltages can be produced at the AC terminal of the inverter. The Fourier analysis of the voltage shows a fundamental component and high frequency harmonics at the carrier and sideband frequencies. The peak of the fundamental component in the region of $0 < m_a < 1$ is found to be goverened by the relation,

$$\hat{v}_{1i} = \frac{m_a V_{dc}}{2} \tag{4.10}$$

This suggest that by varying m_a, the amplitude of the fundamental AC voltage can be regulated and the phase angle can be regulated through the variation of the angle δ.

4.3.1.2 Control Scheme in *d*-*q* Frame

The objective of this control scheme is to determine the required m_a and δ for any particular level of power transfer in such a way that any change in the reference power level will be reflected through a change in these two parameters. This will ultimately change the gating signals and power transfer will be maintained dynamically without any interruption. . The differential equations which describe the AC side of the converter 1 of Fig. 4.1 are as follows;

$$\begin{bmatrix} v_{1a} \\ v_{1b} \\ v_{1c} \end{bmatrix} = R \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \end{bmatrix} + L\frac{d}{dt} \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \end{bmatrix} + \begin{bmatrix} v_{1ia} \\ v_{1ib} \\ v_{1ic} \end{bmatrix} \tag{4.11}$$

where, R and L are the line resistance and inductance, respectively, v_1 is the supply side AC voltage and v_{1i} is the voltage at the terminal of the inverter. Now, using the transformations described in (4.4), (4.6) and 4.7), (4.11) is transformed into d-q frame rotating at $\omega = 2\pi f$,

$$\begin{aligned} v_{1d} &= Ri_{1d} + L\frac{di_{1d}}{dt} - \omega L i_{1_q} + v_{1id} \\ v_{1q} &= Ri_{1_q} + L\frac{di_{1q}}{dt} + \omega L i_{1d} + v_{1iq} \end{aligned} \tag{4.12}$$

or,

$$\begin{aligned} \frac{di_{1d}}{dt} &= -\frac{R}{L}i_{1d} + \omega i_{1_q} + \frac{v_{1d}}{L} - \frac{v_{1id}}{L} \\ \frac{di_{1q}}{dt} &= -\frac{R}{L}i_{1q} - \omega i_{1d} + \frac{v_{1q}}{L} - \frac{v_{1iq}}{L} \end{aligned} \tag{4.13}$$

Rewriting (4.12), and using $v_{1d} = |v_1|$ and $v_{1q} = 0$ for a balance system,

$$\begin{aligned} v_{1id} &= -Ri_{1d} - L\frac{di_{1d}}{dt} + \omega L i_{1_q} + |v_1| \\ v_{1iq} &= -Ri_{1_q} - L\frac{di_{1q}}{dt} - \omega L i_{1d} \end{aligned} \tag{4.14}$$

As the above equation is cross-coupled between i_{1d} and i_{iq}, let us introduce two new variables, x_1 and x_2, so that (4.14) becomes

$$\begin{aligned} v_{1id} &= L(\omega i_{1_q} - x_1) + |v_1| \\ v_{1iq} &= L(-\omega i_{1d} - x_2) \end{aligned} \tag{4.15}$$

Now, using (4.15), (4.13) can be re-written as

$$\begin{aligned} \frac{di_{1d}}{dt} &= -\frac{R}{L}i_{1d} + x_1 \\ \frac{di_{1q}}{dt} &= -\frac{R}{L}i_{1q} + x_2 \end{aligned} \tag{4.16}$$

which are two simple first order equations for i_{1d} and i_{1q} that allows independent control of them. The two newly introduced variables are in fact the two outputs from two PI controllers and can be defined as

$$x_1 = k_d^p(i_{1d}^* - i_{1d}) + k_d^i \int_0^\tau (i_{1d}^* - i_{1d})d\tau \tag{4.17}$$

$$x_2 = k_q^p(i_{1q}^* - i_{1q}) + k_q^i \int_0^\tau (i_{1q}^* - i_{1q})d\tau \tag{4.18}$$

or in the frequency domain,

$$X_1(s) = \left(k_d^p + \frac{k_d^i}{s}\right)(I_d^* - I_d) \tag{4.19}$$

$$X_2(s) = \left(k_q^p + \frac{k_q^i}{s}\right)(I_q^* - I_q) \tag{4.20}$$

For DC side, because of the back-to-back system the capacitors of the two converters are accumulated together and the capacitor current is given as

$$C\frac{dV_{dc}}{dt} = i_{dc1} + i_{dc2} \tag{4.21}$$

where C is the total capacitor of the DC-link, i_{dc1} is the current coming from the converter 1 to the DC-link and i_{dc2} is the current coming out from converter 2 to the DC-link. One of these two currents will have negative sign indicating that the power is transferred from one converter to the other. Neglecting losses, the total AC power sent from the converter must be equal to the DC power at any point of the DC link, therefore,

$$P_{dc} = V_{dc}i_{dc1} = V_{dc}i_{dc1} = P_{ac} = v_a i_a + v_b i_b + v_c i_c \tag{4.22}$$

Using the assumption of balanced AC system and expressing in d-q form,

$$i_{dc1} = \frac{3|v_1|i_{1d}}{2V_{dc}} \tag{4.23}$$

$$i_{dc2} = \frac{3|v_2|i_{2d}}{2V_{dc}} \tag{4.24}$$

Hence, (4.21) becomes,

$$\frac{dV_{dc}}{dt} = \frac{1}{C}\left[\frac{3|v_1|i_{1d}}{2V_{dc}} + \frac{3|v_2|i_{2d}}{2V_{dc}}\right] \tag{4.25}$$

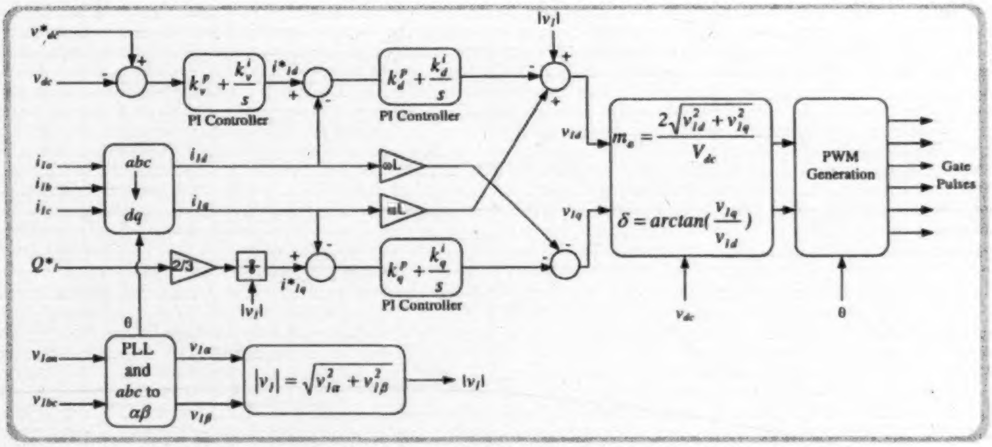


Figure 4.6: Decoupled control scheme for converter 1.

This relationship indicates that the DC voltage can be controlled by controlling i_d and, therefore, a PI compensator is used to produce the reference i^*_{1d} for converter 1,

$$I^*_{1d}(s) = \left(k^p_v + \frac{k^i_v}{s}\right)(V^*_{dc} - V_{dc}) \tag{4.26}$$

Even though PI controllers are expressed in the s-domain, all the controllers are implemented in the discrete domain. The design for the controllers are also carried out in the discrete domain and described in the next section.

The control scheme thus uses two inner current controller for i_{1d} and i_{1q} where the reference signals for these two controllers are supplied by one outer controller and one linear relationship between reactive power, Q_1 and i_{1q} ($Q_1 = 3|v1|i_{1q}/2$). It can be mentioned that the response of the inner controller is much faster whereas the outer controller response is slower due to the presence of large capacitance. The decoupled control scheme for converter 1 is given in Fig. 4.6. An alternate to using m_a and δ to generate reference waveforms could be the use of reference-frame theory ($dq0 \longrightarrow \alpha\beta \longrightarrow abc$) to generate reference waveforms directly which can then be fed for PWM generation.

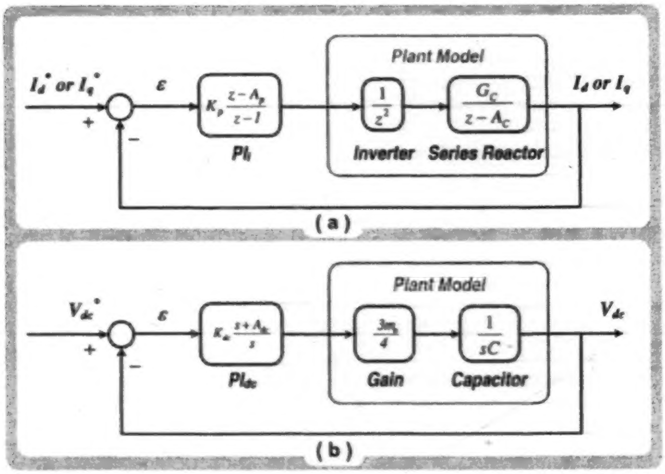


Figure 4.7: Control loops (a) Inner loop PI controller for i_d and i_q (b) Outer loop PI controller for V_{dc}.

4.3.1.3 Design of PI controllers

The output of converter 1 is influenced by three PI controllers. Two of them are used for inner loop, controlling the current, i_d and i_q, and the third one is used for outer loop, controlling the DC-link voltage, V_{dc}. The closed loop diagram for both type of PI controllers is given in Fig. 4.7. The VSC converter was modeled by a delay of two sample periods [74]. From (4.12), the transfer function of the series reactor is given as:

$$G(s) = \frac{I_d(s)}{V_d(s)} = \frac{I_q(s)}{V_q(s)} = \frac{1}{R + sL} \tag{4.27}$$

The above equation is in time domain and must be approximated to discrete form. Step-invariant transformation is used to discretize the above function and the discretized transfer function is found as:

$$\begin{aligned} G(z) &= \frac{1}{R}\left[\frac{1 - e^{-\frac{R\Delta t}{L}}}{z - e^{-\frac{R\Delta t}{L}}}\right] && (4.28) \\ &= \frac{G_c}{z - A_c} && (4.29) \end{aligned}$$

where, $G_c = \frac{1}{R}[1 - e^{-\frac{R\Delta t}{L}}]$, $A_c = e^{-\frac{R\Delta t}{L}}$ and Δt is the discretizing time-step. The transfer function of the PI controller in s-domain is given as:

$$G_{pi}(s) = k_p + \frac{k_i}{s} \tag{4.30}$$

Converting that into z-domain using step invariant transformation:

$$G_{pi}(z) = k_p \frac{z - A_p}{z - 1} \tag{4.31}$$

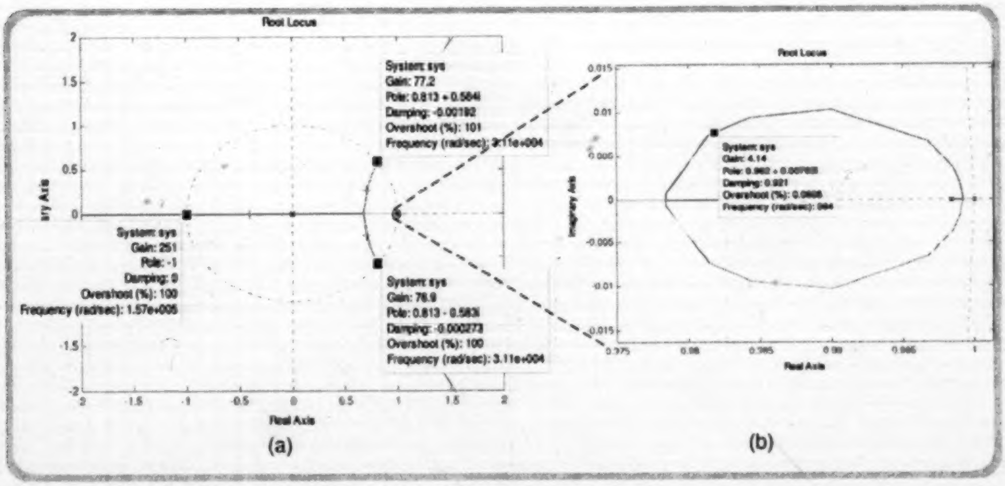
where, $A_p = \Delta t \frac{k_i}{k_p}$.

The DC-link capacitor is ignored since its time constant is much larger than the time step ($20\mu s$) used in the simulation. Having the simplified control loop disctretized, a MATLAB program is written to perform the root-locus design. By looking at the transfer functions, a total of four *poles* are found including a double *pole* at the origin. There is only one *zero* at A_p. For the given values of R and L, it was found that A_c is close to 1 and therefore, A_p should be closer to 1 to compensate the pole located at the edge of the unit-circle in the z-domain. The plot of root locus is given in Fig. 4.8(a), where boundary points are shown giving the maximum limit of k_p. A value of $A_p = 0.989$ is chosen for this root locus. Keeping the boundaries in mind and considering the damping and overshoot within the desired limit, a gain of $k_p = 4.14$ is chosen which is shown in the zoomed view in Fig. 4.8(b). The value of k_i is then calculated from A_p and k_p. A fine tuning of k_p and k_i may be necessary for desired response.

For designing the outer loop PI controller that maintains the DC-link voltage, the series impedances in the AC side is neglected for its very low time-constant with respect to the DC-link capacitance. The large capacitance ($6600\mu F$) would have a very slow response and therefore, the design of the PI controller can be done in the continuous domain (s-domain) [74]. The transfer function of the plant involving DC-link is thus can be given as:

$$\frac{Id(s)}{Vdc(s)} = \frac{3m_a}{4Cs} \tag{4.32}$$

In the control loop diagram shown in Fig. 4.7(b), PI controller transfer function is rearranged to suit the design by writing $k_{dc} = k_i$ and $A_{dc} = k_i/k_p$. The modulation



(a) (b)

Figure 4.8: PI controller design: (a) Root locus including boundary limits (b) Zoomed view of the selected gain point.

index, m_a is assigned a typical value of 0.8 for the design purpose. By inspection, it is found that the overall transfer function has two *poles* at the origin. The *zero* created by A_{dc} should therefore be closer to the origin. The objective behind choosing A_{dc} closer to the *poles* is to direct the *poles* toward stable zone in the s-domain. After a few trial and error and subsequently verifying the values of k_i and k_p in the simulation, they are chosen as $k_p = 2.8$ and $k_i = 200$. The root-locus plot and the response of the system is given in Fig. 4.9.

4.3.2 Control of Converter 2

The control scheme for converter 2 is exactly similar to the control scheme of controller 1 with a little modification in deriving ${i_{2d}}^*$. The outer controller that used to control the DC voltage in controller 1, will now be replaced by a linear equation which will deliver the direct axis reference current ${i_{2d}}^*$. Therefore, the equations used for the reference value of ${i_{2d}}^*$ and ${i_{2q}}^*$ are:

$$ {i_{2d}}^* = \frac{2P_2^*}{3|v_2|} \tag{4.33} $$

$$ {i_{2q}}^* = \frac{2Q_2^*}{3|v_2|} \tag{4.34} $$

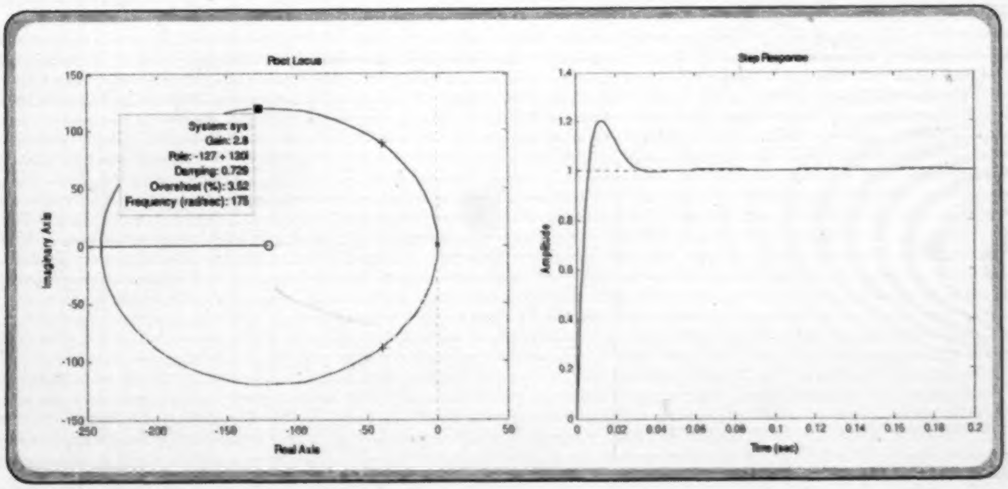


Figure 4.9: PI controller design for DC-link: (a) Root locus plot (b) Step response.

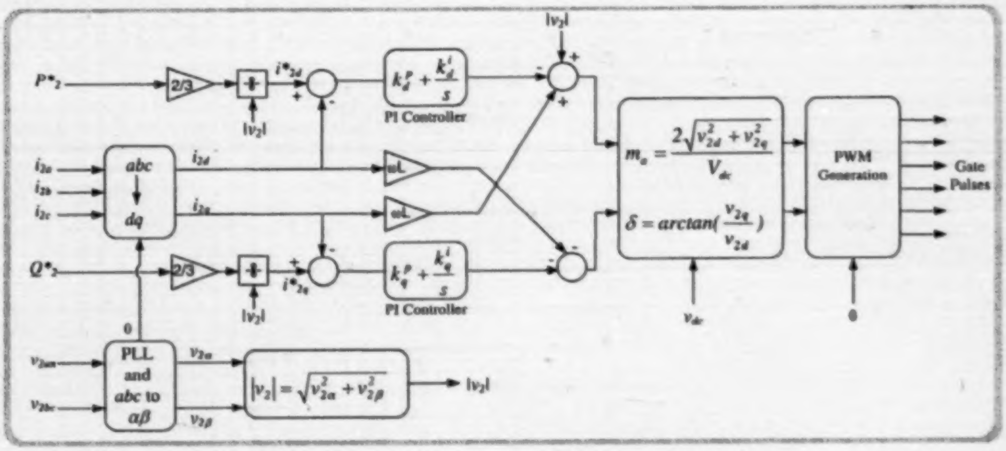


Figure 4.10: Control scheme for converter 2.

where, P_2^* and Q_2^* are the reference power or demanded power at converter 2. The control scheme for converter 2 is shown in Fig. 4.10. Similar design procedure is followed and similar gains of controller 1 are used for the two PI controllers used in the inner loop of controller 2.

4.4 Modeling of VSC-HVDC for Off-line Simulation

Off-line simulation of VSC-HVDC is carried out by using PSCAD/EMTDC and MATLAB/SIMULINK. Both programs are powerful time-domain transient simulators and are capable of simulating power systems, power electronics and their controls. Both uses graphical user interface to construct virtually any electrical/electro-mechanical network and provide a fast and flexible solution. PSCAD/EMTDC represents and solves the differential equations of the entire power system and its control in the time domain (both electromagnetic and electromechanical systems) [47]. It employs the well-known Nodal analysis technique together with Trapezoidal integration rule with a fixed integration time-step. It also uses interpolation technique with instantaneous switching to represent the structural changes of the system.

MATLAB/SIMULINK is a high-performance multi-functional software that uses functions for numerical computation, system simulation, and application development. Among many other toolboxes, Power System Blockset (PSB) under the name of *SymPowerSystem* is one of its design toolboxes for modeling and simulating electric power systems within the SIMULINK environment [71]. It contains a block library with common components and devices found in electrical power networks that are based on electromagnetic and electromechanical equations. PSB/SIMULINK can be used for modeling and simulation of both power and control systems. PSB solves the system equations through state-variable analysis using either fixed or variable integration time-step. The linear dynamics of the system are expressed through continuous or discrete time-domain state-space equations. It also offers the flexibility of choosing from a variety of integration algorithms.

4.4.1 Modeling of VSC-HVDC System in PSCAD/EMTDC

The power circuit of VSC-HVDC system is modeled using PSCAD/EMTDC v 4.2.1. The complete model of the system and its controls developed in PSCAD/EMTDC is given in Appendix B.

A. AC Voltage Source

The two AC supply sources for both converter 1 and converter 2 sides are represented through a 3 phase AC voltage source whose parameters can be adjusted. The sources are modeled as grounded Y connected with a frequency of 60 Hz and rms magnitude of 117V.

B. Phase Reactors

Phase reactors for both sides are modeled using an inductor of 2.5mH in series with a series resistance of 0.2 Ω. The series resistance accounts for the internal resistance of the inductor as well as the resistance offered by the cables connecting them.

C. Converters

Six individual IGBTs are used to construct the model of the converter. The gating signals of the converters are generated and supplied by the control algorithm. Snubber circuits are excluded from the model as it simplifies the simulation to a great extent with the exchange of losing some accuracy. The IGBTs are modeled with the parameters as shown in Table 4.1.

Table 4.1: IGBT parameters used for the simulation.

Parameters	Values
Snubber resistance	1e9
Snubber capacitance	∞
Switch ON resistance	0.001Ω
Switch OFF resistance	1e6Ω
Device forward voltage	0V
Fall time	1e-6s
Tail time	2e-6s

D. DC-link

The test system is a back-to-back HVDC link and therefore, there is no DC transmission line between the converters. A capacitor is modeled with the ratings of 6600μF, 900V. A few discharging resistors with a total value of 9k Ω are also added in the system model.

E. Controllers

Controllers for both converter 1 and converter 2 are modeled using the blocks available in the PSCAD/EMTDC library module. Custom blocks/functions are also used when there is no built-in block present in the library. The control model mainly consists of feedback measurements of currents and DC voltage, and generation of firing signals. The Phase Locked Loop (PLL) is used to synchronize the firing signals. The output of PLL is a ramp, synchronized to the phase A bus line-to-ground voltage, which is used to generate the reference signals. PWM signals are then generated by comparing the triangular wave with these reference signals. The frequency of the triangular carrier signal is chosen as $2kHz$ The detailed model of the controllers are given in Appendix B.

4.4.2 Modeling of VSC-HVDC System in MATLAB/SIMULINK

SIMULINK version 6.1 along with SimPowerSystem version 4.0 has been used to model the VSC-HVDC system. The state-space representation of the linear electrical circuit of the system produced 6 states, 15 inputs, 27 outputs and 12 switches. The detailed model is given in Appendix B. A brief description on the modeling of the system is given here.

A. AC Voltage Source

Three single-phase AC voltage sources are connected in Y with a ground connection to make the sources for both converters. The source parameters are kept the same as it is used in PSCAD/EMTDC simulation.

B. Phase Reactors

A three phase series reactor of $2.5mH$ with a resistance of 0.2Ω in series are added on both sides of the system. The reactor block connects converters to the system. The parameters are kept at the same value as it is used in PSCAD/EMTDC modeling.

C. Converters

The two converters are modeled with the two *Universal Bridge Blocks.* The *Universal Bridge Block (UBB)* implements a universal three-phase power converter that consists of six power switches connected as a bridge. The type of power switch and converter configuration can be selected from the dialog box and IGBT is chosen in this case. Series RC snubber circuits are connected in parallel with each switch device, however, they are eliminated by chosing very high value as shown in Table 4.1. The gating signals are 6-pulse trains for each converter and therefore, six gating signals should be sent to the UBB converter.

D. DC-link

Similar to the PSCAD model, DC-link is modeled using single capacitor with the same rating of 6600 μF, 900V. A discharging resistor of 9k is also modeled in parallel with the capacitor.

E. Controllers

The control blocks available in SIMULINK library have been used to implement the control scheme depicted in Fig. 4.6 and Fig. 4.10, and enough care has been taken to ensure that exact parameters as in PSCAD/EMTDC simulation are used. Feedback signals such as v_{abc} and i_{abc} are obtained using 3-phase V-I measurement blocks. The discrete gate signals generated by the PWM block are fed to the Universal Bridge Block.

4.5 Off-line Simulation Results

The off-line simulation, using these two well-known simulation programs is carried out with an objective of validating the operation of VSC-HVDC system and its control design. In addition to that a comparative study between the two programs will benefit in understanding the advantages and limitations of both programs. Steady-state and transient results (created through various faults) are recorded and then compared.

The comparison reveals high degree of similarity among the results obtained through both simulation environments with minor discrepancies.

4.5.1 Steady-State

For steady-state analysis, the system has been simulated for a few seconds, however, results are shown for shorter periods due to the large amount of data as well as the clarity of the figures. There were some initial transients which subsided very fast and the system reached steady-state thereafter. Two cases have been considered and for each case, results are plotted for DC voltage, AC currents, line-to-line voltage at the terminals of both converters, phase voltages and power levels. The time-step used for the entire simulation is $20\mu s$ under both simulation environments. Closed-loop simulation has been carried out as it is not possible to transfer power between two grids in an open-loop VSC-HVDC system.

Case I: All the reference power are set to positive values

In this case, the reference quantities are set as:

- $P_2 = 4000\,W$ and $Q_2 = 2000\,VAr$ (inductive)
- $Q_1 = 2000\,VAr$ (inductive) and $V_{dc} = 500$ V

Fig. 4.11 shows line-to-neutral voltages, line-to-line voltages, phase-a currents and their harmonic spectrums obtained through both PSCAD/EMTDC and PSB/SIMULINK. The left column represents the results obtained through PSCAD/EMTDC and the right column is the results obtained through PSB/SIMULINK. It can be concluded that both programs produce almost identical results. The fundamental component of current has a magnitude of $18.08A$ and the current has a THD of 21.88% in PSCAD/EMTDC, whereas PSB/SIMULINK shows the magnitude of fundamental component as $18.42A$ and the THD as 19.92%. Similar results have been observed in case of the voltages and currents in converter 2. Fig. 4.12 shows the results for voltages and currents of converter 2, obtained by both simulation programs. Identical results are observed for both programs and harmonic analysis of the currents shows that the fundamental components and harmonics are

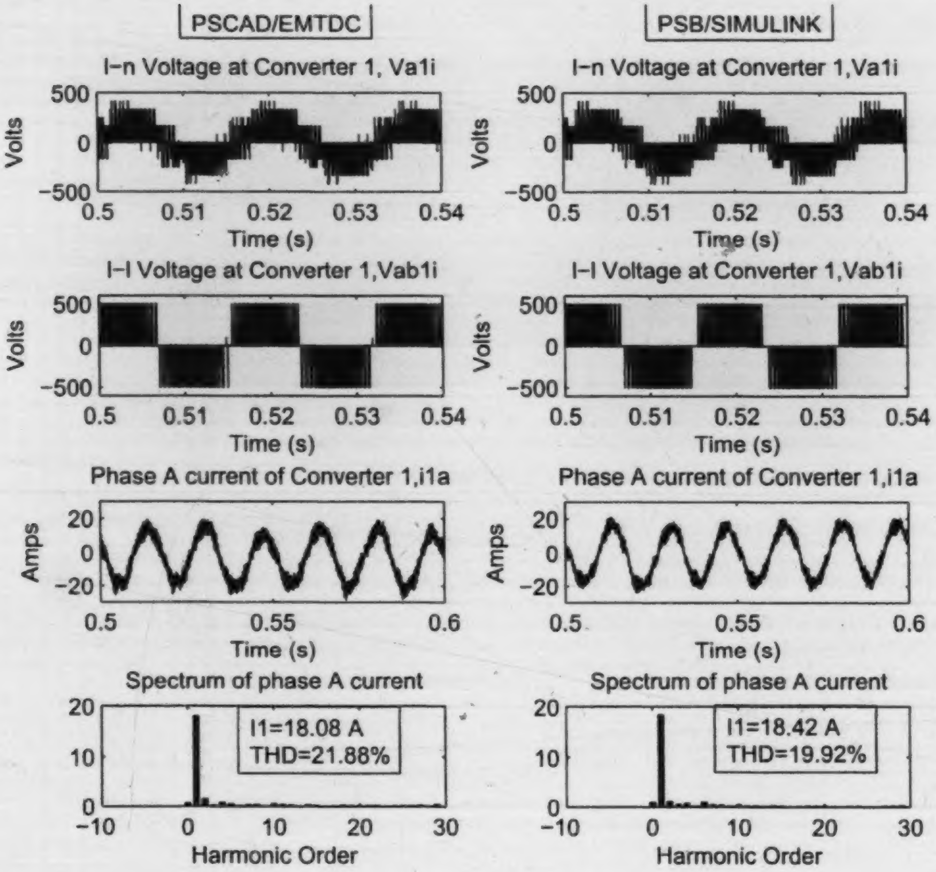


Figure 4.11: Voltages and currents at converter 1 using both PSCAD/EMTDC and PSB/SIMULINK.

also in closer proximity. The $60Hz$ fundamental components are $18.75A$ (PSCAD) and $19.61A$ (SIMULINK), and THDs are 20.92% (PSCAD) and 16.78% (SIMULINK).

DC-link voltage and power levels at both converters are shown in Fig. 4.13. At a glance, the two simulation programs produce similar results for all waveforms, neglecting minor discrepancies during the initialization. For DC-link voltage, V_{dc}, the initial transient is slightly higher in PSCAD/EMTDC than V_{dc} obtained in PSB/SIMULINK. However, within a very short time, V_{dc} settles to its reference value

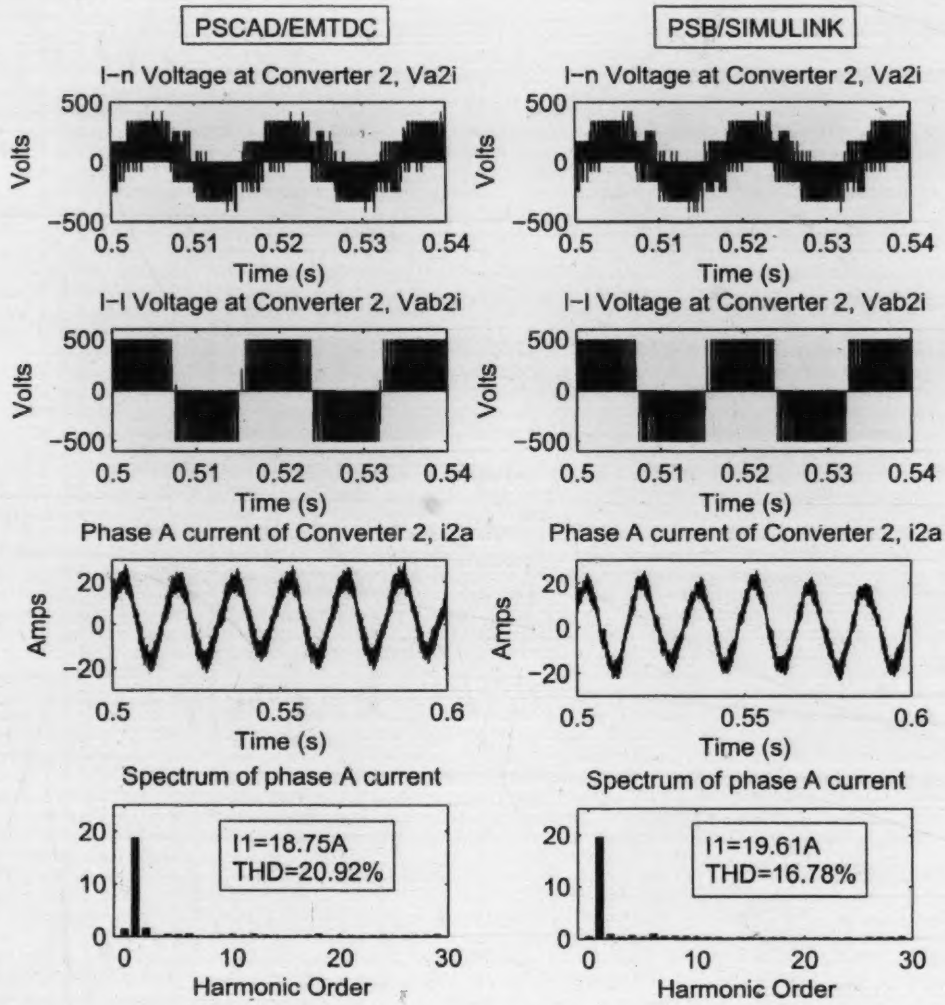


Figure 4.12: Voltages and currents at converter 2 using both PSCAD/EMTDC and PSB/SIMULINK.

Of $500V$.

Similar to DC-link voltage, an initial transient is observed in power levels which die down after a while and the steady-state power level is reached. Neglecting minor oscillations present in the power levels at steady-state, the power levels are found to be equal to their corresponding reference values. As reactive powers are generated

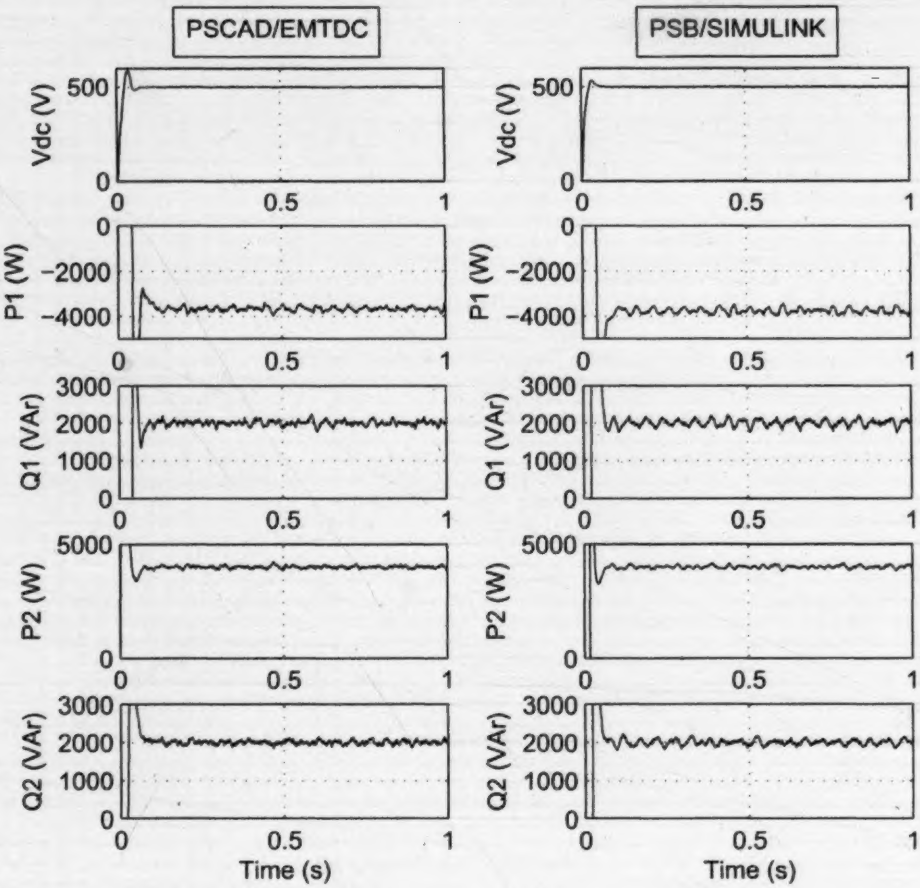


Figure 4.13: DC voltages and powers using both PSCAD/EMTDC and PSB/SIMULINK.

locally, they are exactly equal to their reference quantities (2000 VAr), however, active power is transmitted from one converter to the other through the DC-link. As a result, there will be some power loss in the converters as well as in the dc-link discharging resistors. The active power flow is considered positive when it flows to the DC-link from the grid-line and as shown in the controller scheme of converter 2, the power reference is maintained at converter 2. Therefore, if reference active power, P_2 is set to a positive quantity, the power P_1 will be negative. This means, the power flow in converter 1 takes place from DC-link to the grid line and that is exactly what can be seen in Fig. 4.13. While P_2 is found to be at $4000W$ (which is equal to the reference value), the average value of P_1 is found approximately $-3678.3W$ that indicates power flow from converter 1 to the grid-line and a loss of $321.7W$ in the transmission process.

Case II: All the reference powers are set to negative values

For this case, active power demand is reversed and reactive power reference is also changed to negative while DC voltage reference remains unchanged. The reference quantities are set as:

- $P_2 = -4000W$ and $Q_2 = -2000\,VAr$ (capacitive)
- $Q_1 = -2000\,VAr$ (capacitive) and $V_{dc} = 500V$

Similar to case 1, almost identical results are observed in this case. The steady-state voltages and currents at converter 1 are shown in Fig. 4.14. The fundamental component of currents are found to be $18.25A$ in PSCAD and $18.74A$ in SIMULINK. THDs are also very close; 21.75% and 17.89%. Voltages and currents found at the terminal of converter 2 are also plotted in Fig. 4.15. Voltage and currents are found to be almost same in both programs. The fundamental currents and THDS are $16.99A$ and 22.71% in PSCAD whereas in SIMULINK they are found to be $17.66A$ and 20.17%. The PSCAD produces higher THDs due to the presence of stronger DC component in the current.

In Fig. 4.16, DC-link voltage and power levels are plotted for this case. The DC voltage is almost similar to that of Case I. With a slight initial transient the DC-link voltage settles to steady-state value at $500V$. The reactive powers are exactly equal to its reference values ($-2000\,VAr$), although there is a slight oscillation in the power which is due to the presence of harmonics in the voltages and currents. The active power reference was set to $-4000W$ which means, this amount of power should flow to converter 2 from the dc-link. To produce this amount of power from dc-link, converter 1 supplied almost an average of 4361.5W, out of which $361.5W$ is lost in the series resistor, converters and dc link resistors. This loss is slightly higher than that of Case I (321.7W), and it is due to the slightly higher current flow at converter 1. Considering both cases, it can be concluded that active powers can be transferred

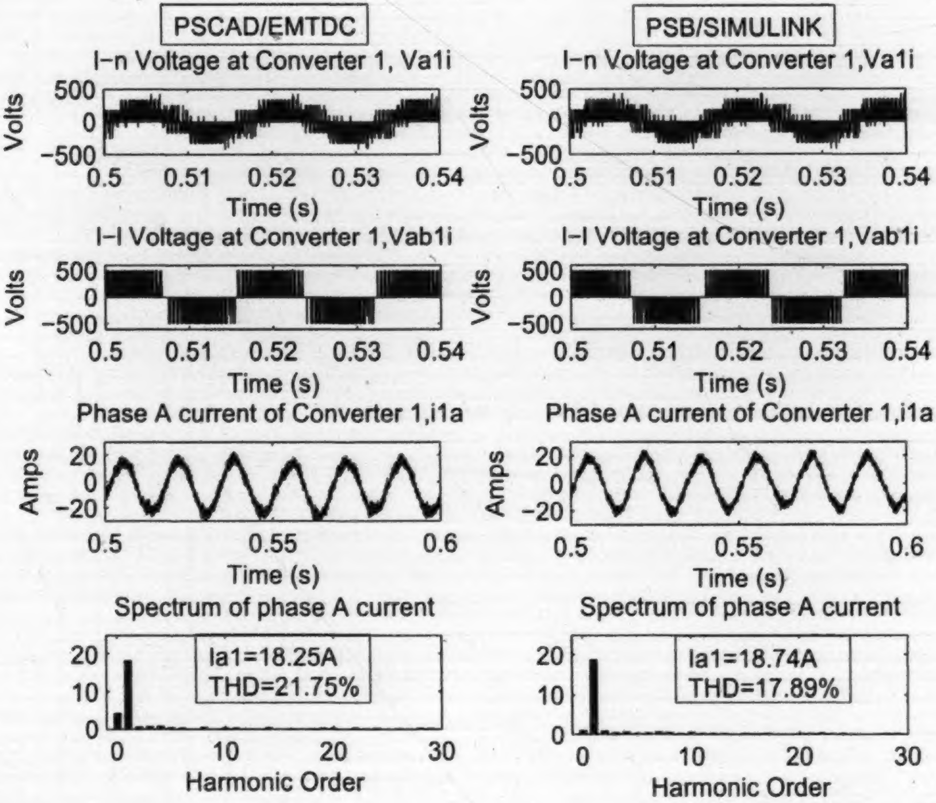


Figure 4.14: Voltages and currents at converter 1 using both PSCAD/EMTDC and PSB/SIMULINK.

between the two AC grids through the VSC-HVDC link and reactive powers can be generated locally.

4.5.2 Transients

A few transient cases have been considered to investigate the response and stability of the system.

Case I: DC voltage reference has been changed while keeping power demand unchanged

Two transitions in the reference DC voltage is created and the responses of various waveforms are observed. While the DC voltage transition is created other reference

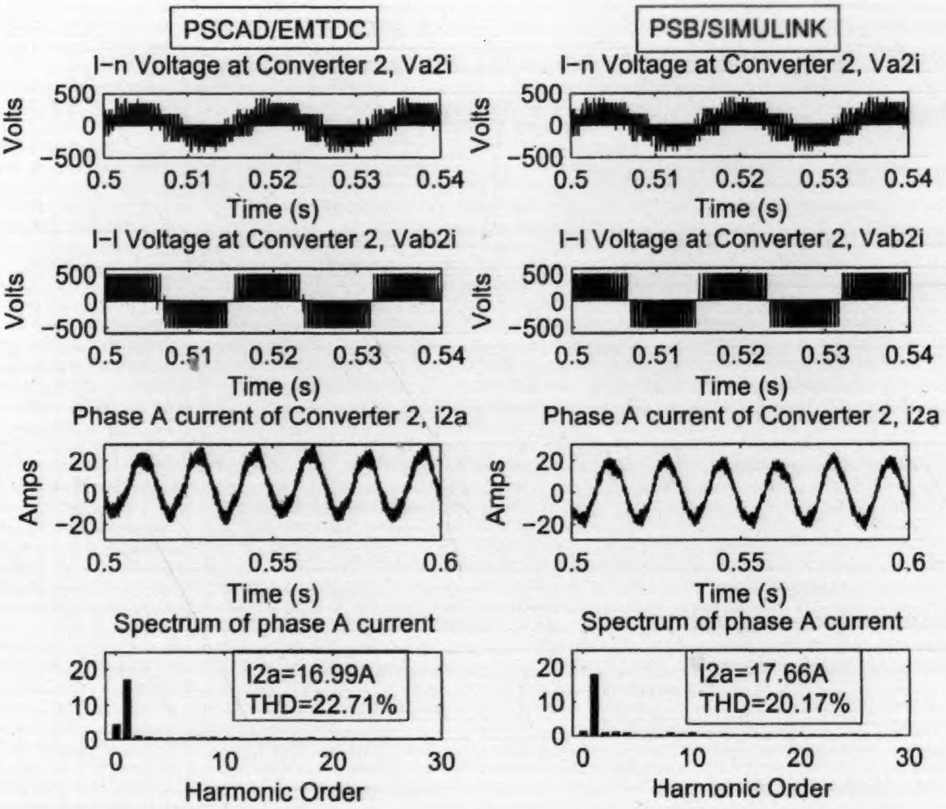


Figure 4.15: Voltages and currents at converter 2 using both PSCAD/EMTDC and PSB/SIMULINK.

quantities are kept unchanged and kept at $P_2 = 4000W$, $Q_2 = 2000\,VAr$ and $Q_1 = 2000\,VAr$. At $t = 0.4$s, the DC voltage reference is changed to 560V and continued until $t = 0.8$s. At $t = 0.8$s, the DC voltage reference is brought down to 440V. Fig. 4.17 shows the transient responses observed in the DC-link voltage and also the responses in the power levels at both converters.

In both transitions, the DC-link voltage is found to follow smooth transitions and then it became stable at its new position. Both PSCAD/EMTDC and PSB/SIMULINK produce almost the same results validating the fact that the designed controller is efficient and stable. Due to this transition of DC-link voltage, momentary disturbances are observed in the power levels of converter 1, however,

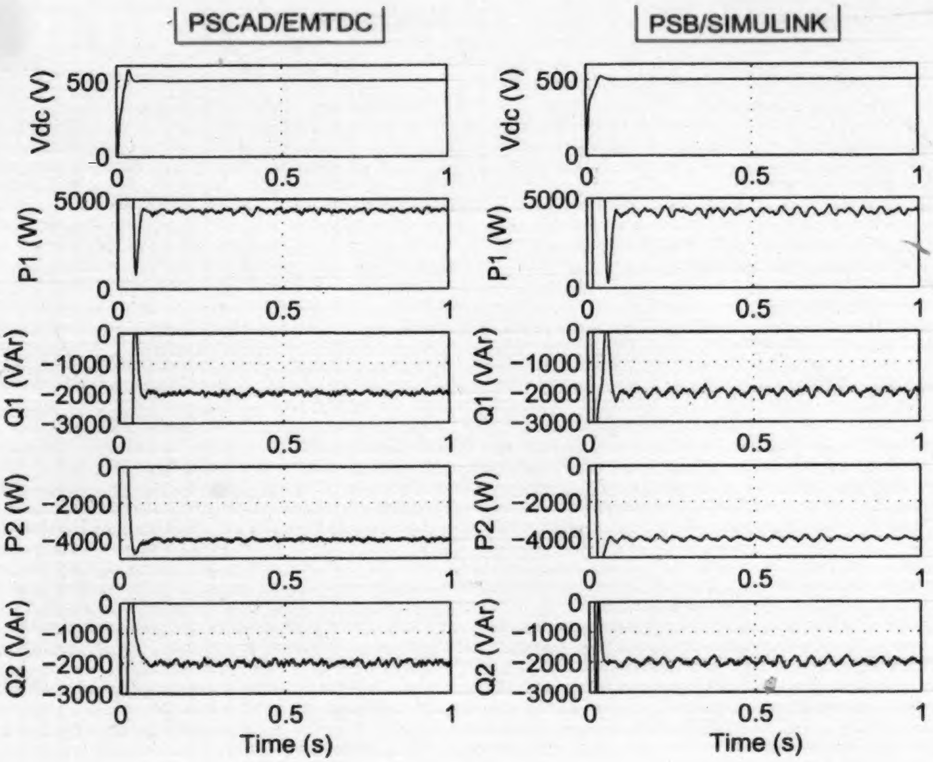


Figure 4.16: DC voltages and powers using both PSCAD/EMTDC and PSB/SIMULINK.

converter 2 power level remains unaffected. This indicates that the DC-link is very strong and as it is controlled by the converter 1, changes in dc-link primarily affect the converter 1. Another important point is that active power is found be more affected than the reactive power due to the disturbance. This is because active power directly depends on the DC-link voltage and reactive power disturbance is due to the changes in the current of converter 1.

Case II: DC-link has been short-circuited for a small period while keeping power demand unchanged

In this case, the DC-link has been short-circuited for 0.05s through a resistor.

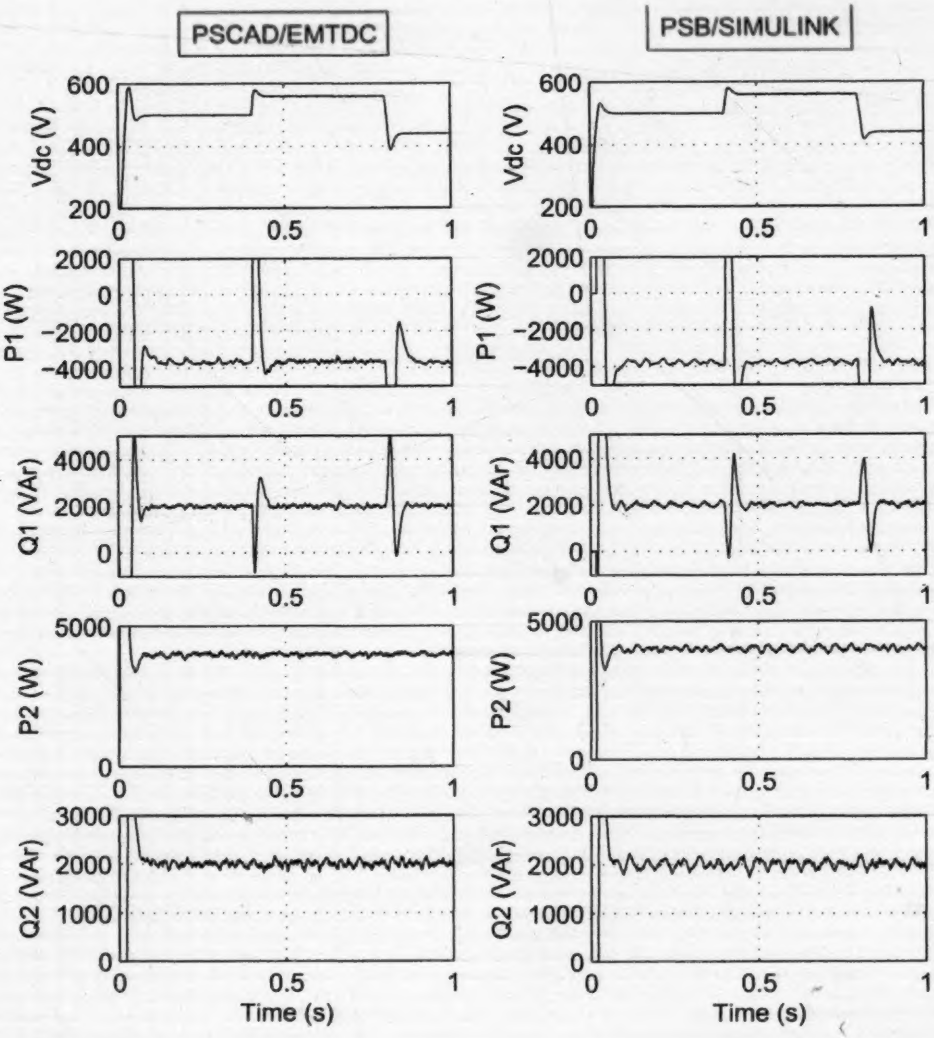


Figure 4.17: DC voltages and powers using both PSCAD/EMTDC and PSB/SIMULINK.

The power references are kept at a setting of $P2 = 4000W$, $Q2 = 2000\,VAr$ and $Q1 = 2000\,VAr$. At $t = 0.4s$, the DC-link is short-circuited through a small resistor of 1Ω and continued until $t = 0.45s$. At $t = 0.45s$, the short-circuit is removed and normal DC voltage is restored in the system. While modeling this in PSCAD/EMTDC a Circuit Breaker (CB) is used which is capable of breaking the contact even there is a current flow through it. However, in SIMULINK, a CB cannot open the contact if there is a current flow through it. It can open the contact only at the zero crossing of the current. Using such a CB was not possible for emulating the DC-link short-circuit situation as the short-circuit current never goes to zero. Therefore, an IGBT is used to make or break the DC-link in SIMULINK. Fig. 4.18 shows the responses of the DC-link voltage and power flow in both converters. When DC-link undergoes a short circuit situation for $0.05s$, the DC-link voltage comes down to a level of approximately $200V$ instead of going down to zero. This is due to the fact that when the DC-link voltage went down, the control scheme could not generate PWM signals and the DC-link capacitor is charged from both sides through the anti-parallel diodes which worked as three phase full-wave rectifiers. As the short-circuit resistance is not zero but very low, a large current flow (approximately $200A$) is observed from both sides of the grid. Because of that, both reactive and active power flow became very high. One interesting phenomenon has been observed in case of active power flow of converter 1. Before the short-circuit situation, as reference power P_2 was positive, P_1 was negative i.e., converter 1 was transferring power to the grid from the DC-link. When the short-circuit happened, the converter 1 power P_1 became positive and instead of supplying power to the grid, it was drawing a large power from the grid. As soon as the short-circuit is cleared, P_1 became negative and normal operation took place. However, the direction of power flow in converter 2 remained unchanged as it was initially supplying power to the DC-link. In case of reactive power flow, during the short-circuit period, both converters became capacitive as they were charging the DC-link capacitors. The waveform of voltages and currents are shown in Fig. 4.19. The short-circuit situation increased the currents in both converters to a high value and the line-to-line voltage is disturbed during that period.

Case III: Power references have been changed while keeping DC-link voltage unchanged

Here, all the power references go through changes as shown in Fig. 4.20 while the DC voltage is kept unchanged at 500V.

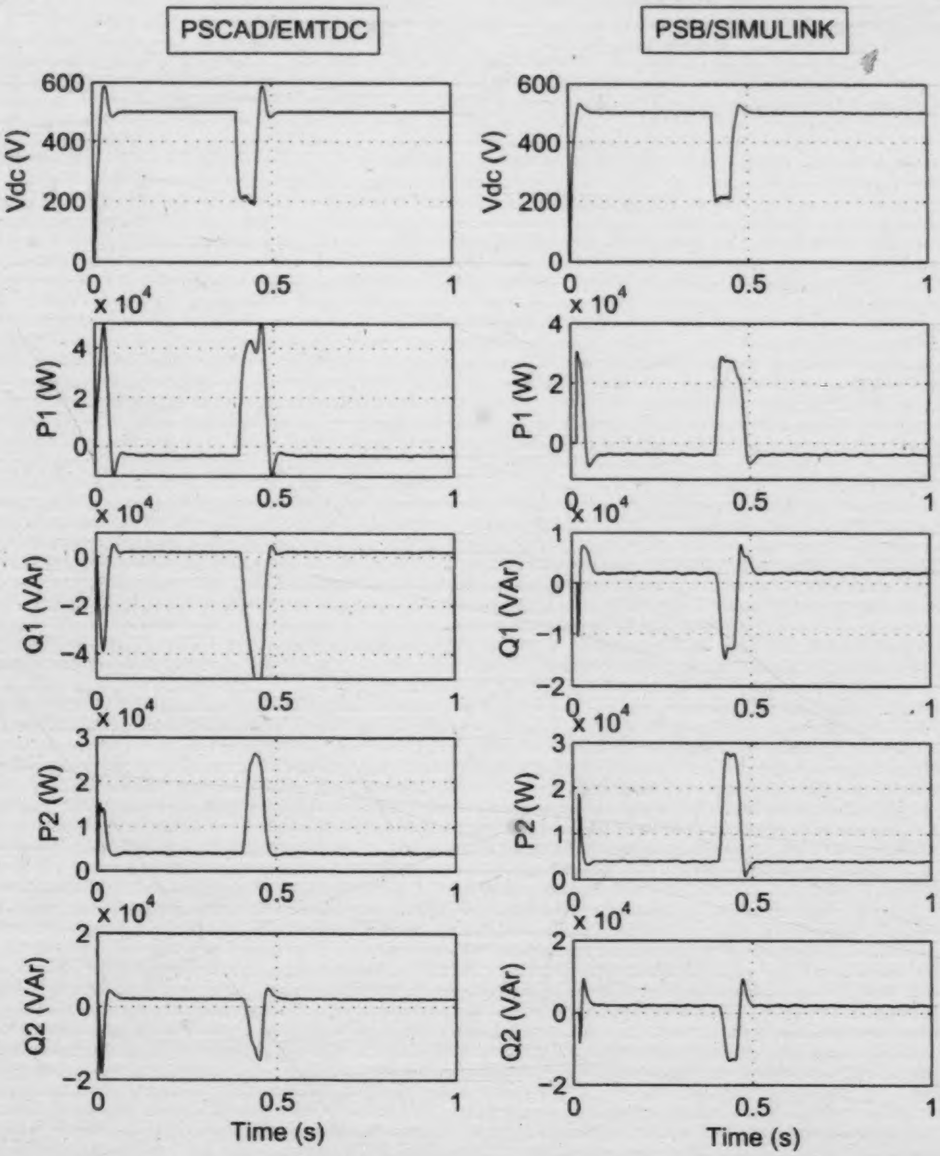


Figure 4.18: Responses of dc-link voltage and power quantities due to a short circuit at dc-link .

Fig. 4.21 shows the transient responses observed. The DC-link voltage seems to be unaffected by the changes in power demands except at $t = 0.7s$, where a little dip in the voltage is observed due to the sudden reversal of real power in the system.

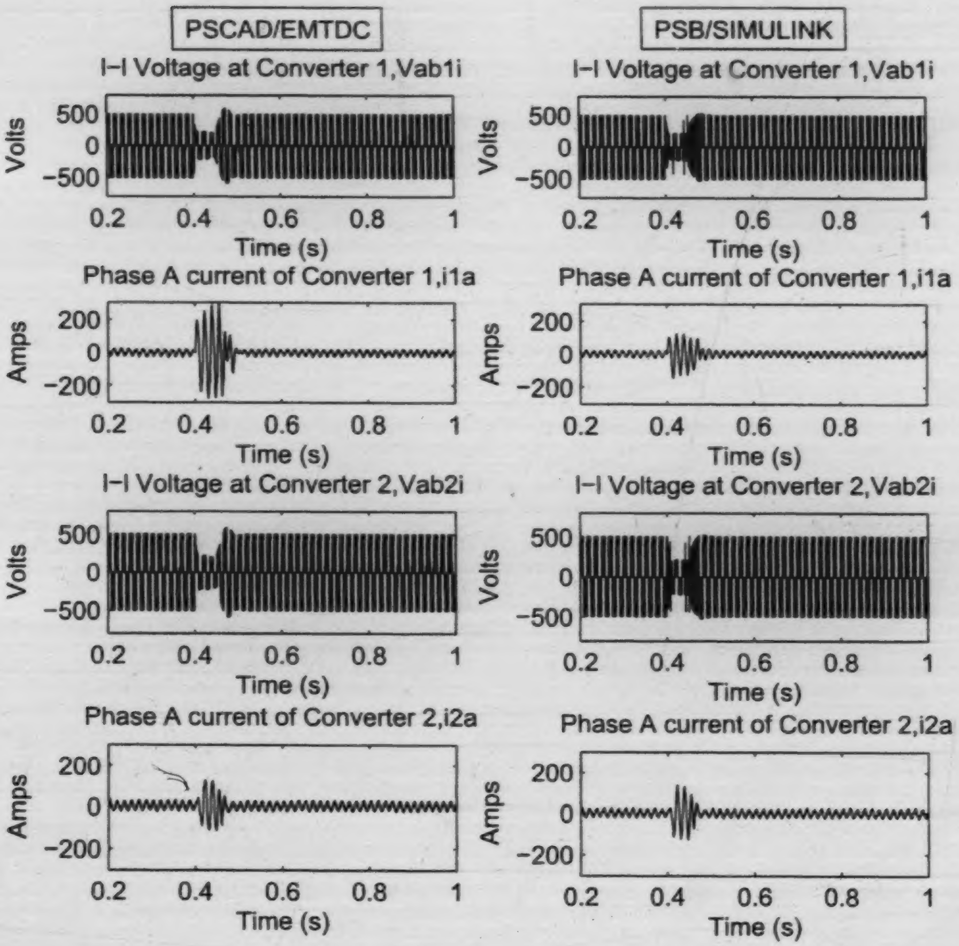


Figure 4.19: Waveforms of voltages and currents while there is a short circuit at dc-link .

The transitions of power levels has taken place smoothly in all cases. Any change on the demand of active or reactive power has created very little or no disturbances on the rest of the power levels. This again corroborates the fact that the controller can handle any change in power demand, even power reversal without losing its stability.

Case IV: AC fault is introduced in the converter 2

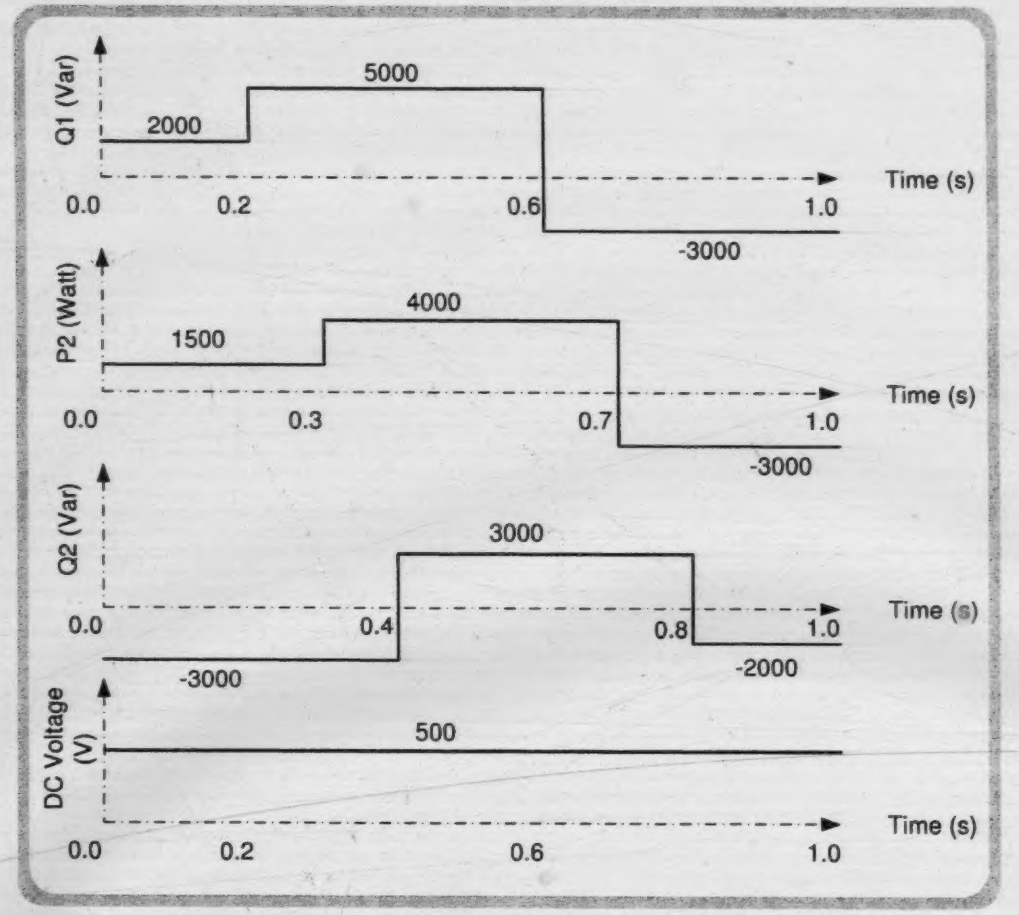


Figure 4.20: Reference values for power and dc-link voltage.

A loss of AC supply to the converter 2 is emulated while the other references are left unchanged. At $t = 0.5s$, the three phase supply to converter 2 is interrupted through the opening of a Circuit Breaker (CB). The supply has been re-instead at $t = 0.6s$. For this study, the reference powers are kept as: $P2 = 4000W$, $Q2 = 3000\,VAr$, $Q1 = -3000\,VAr$ and $V_{dc} = 500V$. The transient responses are observed and shown in Fig. 4.22.

As DC-link voltage is maintained by converter 1, loss of supply to converter 2 does not have any effect on DC-link voltage, however, reestablishment of AC supply

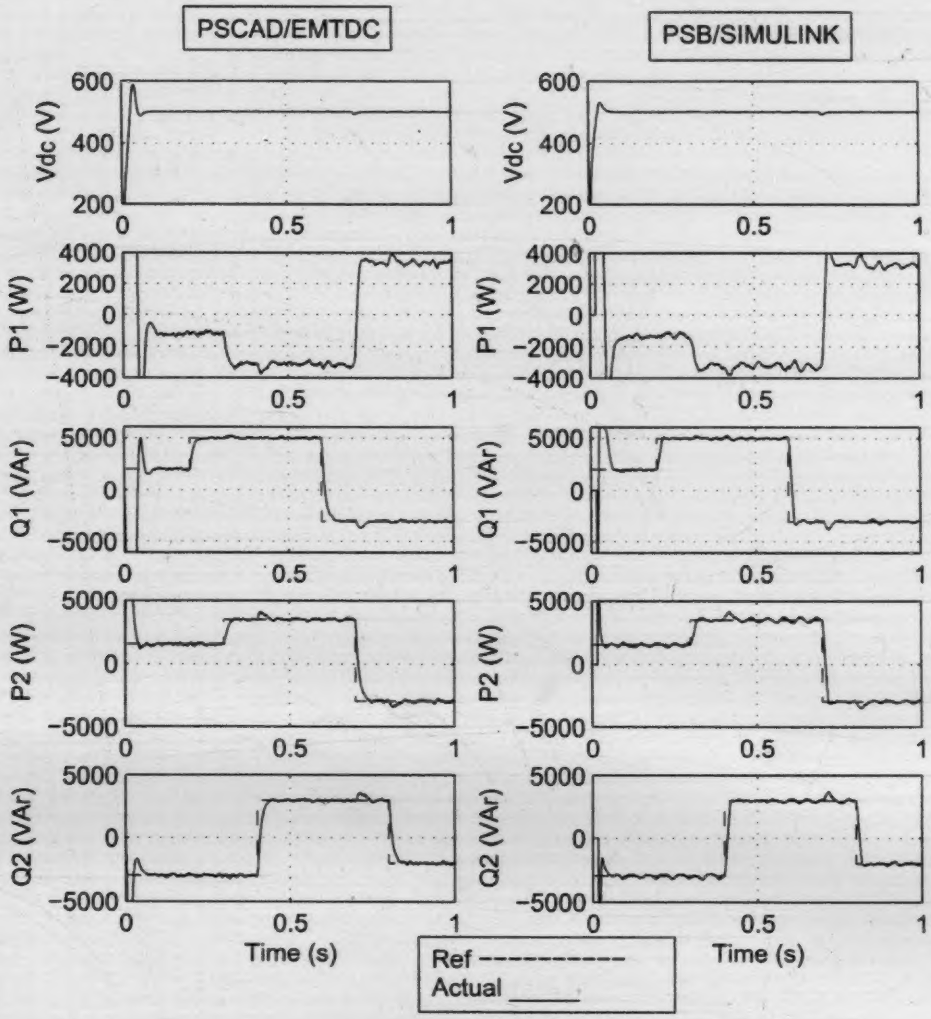


Figure 4.21: Transients in various power levels.

to converter 2 has created a slight disturbance at the dc voltage which is within neglegible limits of error as found in Fig. 4.22. During the black-out period, active power flow in the network, both P_1 and P_2 became zero. This is because active power demand comes from controller 2 which became un-operative due to zero current flow in the converter. As active power demand from converter 2, P_2^* becomes zero, active power supply from converter 1, P_1 is therefore zero (neglecting losses). Similarly, Q_2 also became zero as network connected to converter 2 became isolated, however,

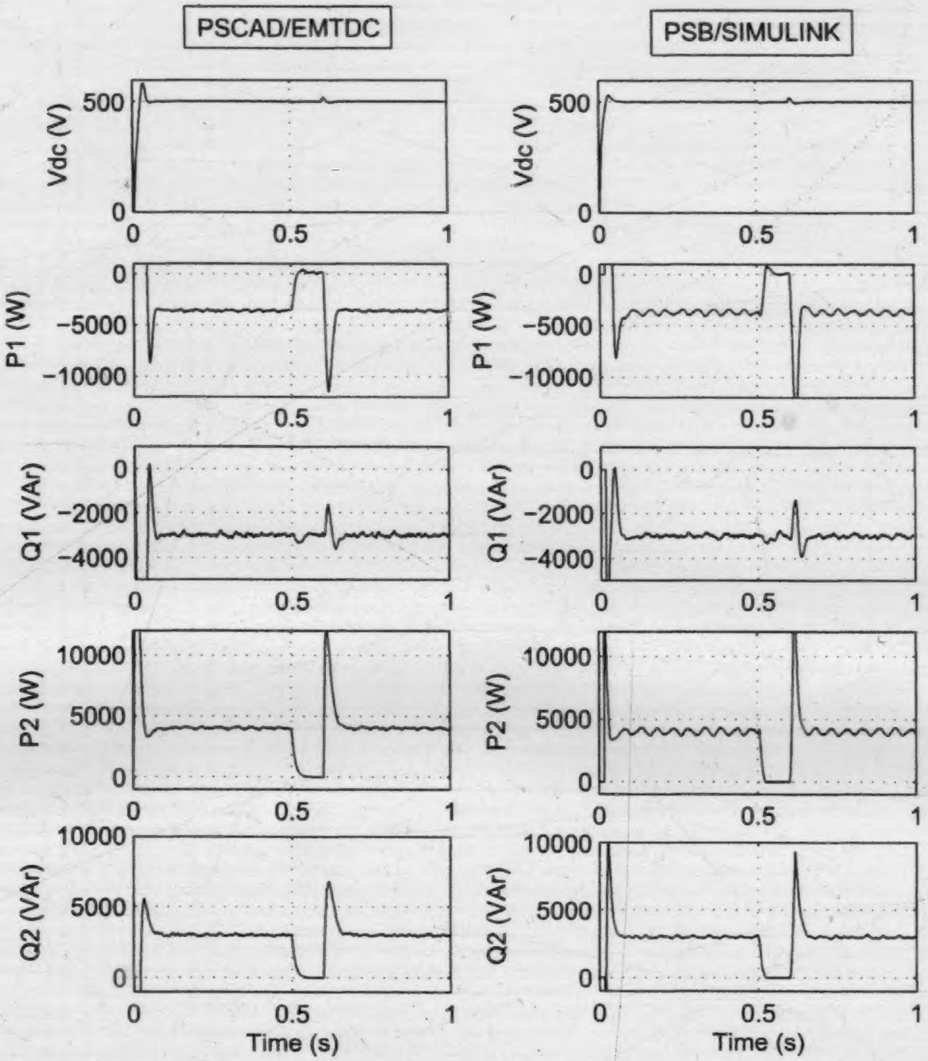


Figure 4.22: Responses of DC-link voltage and powers due to the loss of grid supply to converter 2.

reactive power from converter 1, Q_1 was still flowing. As reactive power is generated locally, Q_1 was supplied by converter 1 and remained almost unaffected despite the network lost its other side.

The voltages and currents at both converters are given in Fig. 4.23. During the isolation of supply voltage at converter 2, voltages and currents at converter 1 are

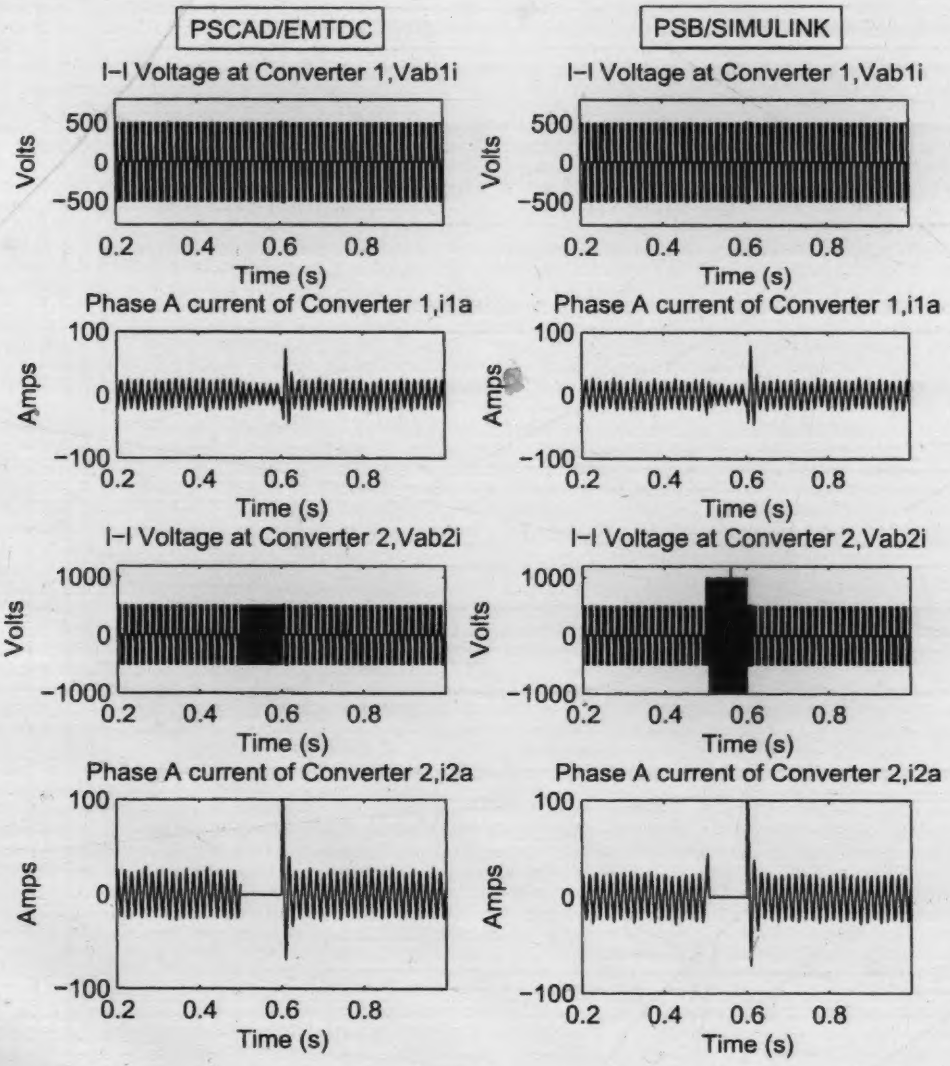


Figure 4.23: Voltages and currents while converter 2 lost its supply from 0.5s to 0.6s .

slightly affected. A decrease in the current is observed while voltage is not apparently affected. This is because converter 1 was functioning smoothly with no active power to supply but to generate reactive power. No demand in real power decreased the current in converter 1 and when demand is re-established, the current reached to its pre-fault value. In converter 2, as the network is isolated, the current became zero and with some transient it went back to its normal value. The nature of transients are same in both PSCAD/EMTDC and PSB/SIMULINK. However, for voltage, during the blackout period, it seems slight difference is found in transient waveforms between results obtained by two programs. This could be due to the inherent deficiencies of the programs and can be validated by comparing with the experimental results.

4.6 Execution Time and Memory

Both programs (PSCAD/EMTDC and PSB/SIMULINK) were run on a Dual Core 2.0 GHz, Intel T2500 processor running Windows XP operating system. MATLAB v. 7.0.1 R 14.1 SP1 with SIMULINK v. 6.1 and SimPowerSystem v. 4.0, and PSCAD/EMTDC v. 4.2.1 were used for simulating the system. The execution time was recorded from the 'Time Summary' shown at the output window in PSCAD/EMTDC, and for PSB/SIMULINK, the function 'cputime' was used at the start and end of the simulation. Information on memory usage was collected from the 'Task Manager'. While these simulations were performed, no other program was allowed to run except the system files. With respect to simulation speed, both programs were found to have almost similar speed. For a simulation duration of 1s, using $20\mu s$ time-step, PSCAD/EMTDC took $7.84s$ with a memory usage of $64.72MB$, whereas PSB/SIMULINK took $6.82s$ and the memory usage was $164.7MB$. Higher time-step reduced the execution time, however, at the cost of reduced accuracy in the simulation results.

4.7 General Remarks

Both PSCAD/EMTDC and PSB/SIMULINK provide user-friendly graphical environments for designing the circuits through simple functional blocks. However, the following minor differences particular to this case study, are worth mentioning:

- PSCAD/EMTDC is a specialized software designed mainly for transient analysis of electrical systems therefore, it has some added advantages such as built-in blocks for PLO based firing control and the measurement of alpha/gamma angles embedded inside the *6-pulse Garetz bridge.* On the other hand, PSB/SIMULINK requires measurement systems to be developed by the user.

- PSB/SIMULINK offers more flexibility in terms of choice of the solution techniques; fixed or variable time step based solutions, discrete or continuous solutions and higher order algorithms for solution techniques. On the other hand PSCAD/EMTDC does not give any other solver options except Trapezoidal Rule. For this study, a fixed step-size, Trapezoidal Rule has been used in PSB/SIMULINK to be consistent with PSCAD/EMTDC.

- PSB/SIMULINK library is very rich in comparison to PSCAD/SIMULINK. Statistical data processing blocks such as such as Mean, Average, Mean Absolute Deviation (MAD) and others are not found in PSCAD/SIMULINK, whereas PSB/SIMULINK offers most of these blocks.

- Sometimes, error debugging system in both programs is quite complex. Instead of locating the exact source of errors, programs just return general error messages. The error debugging needs to be a little more user-friendly in those cases.

4.8 Summary

The control of VSC-HVDC is very important for the smooth operation of a VSC-HVDC system. Desired responses of the system transients are mostly determined by the control algorithms and its implementation. Following are the contributions of this chapter:

- A differential equation based mathematical model is used to develop the control scheme for both VSCs so that they can transfer real-powers from one side to the other or they can generate required reactive power as needed. The design of PI controllers were carried out using step-invariant transformation.

- The VSC-HVDC and its control are then modeled using two well-known transient simulation packages, PSCAD/EMTDC and PSB/SIMULINK.

- Simulation of the system using both programs are carried out for steady-state and transient situations. Sudden changes in the reference DC-link voltage, active power demand and reactive power demand, and faults (DC-link fault and AC supply blackout) are simulated.

- A detailed comparison of performances between the two simulation environments is presented. Interestingly, both programs have produced identical results during steady-state and transients, which validates the accuracy of the modeling, design of the control and simulation algorithms. As there is no major differences in results, both programs can be used for transient analysis of electrical systems.

Chapter 5

Real-Time HIL Simulation of VSC-HVDC and Experimental Validation

5.1 Introduction

This chapter[1] presents the modeling, implementation of real-time HIL simulation and experimental validation of the VSC-HVDC. An off-line simulation is also carried out using the C programming language. The objective of another off-line simulation is to verify the accuracy of the real-time model of the VSC-HVDC system that is used later for HIL-simulation. Discretization, switching synchronization and the control scheme described in the previous chapters are applied here to carry out the HIL simulation. The simulation results are verified through experimental results. For both HIL-simulation and experimental validation, a PC-cluster installed in the Real-Time eXperimental Lab (RTX-Lab) is used. Similar control scheme described in Chapter 4 is used for experimental validation of the simulation results. The digital control scheme is implemented in one of the *Target* nodes of the PC-cluster which can produce $(0-15V)$ gate pulses for both the HIL simulation and the experiment.

---

[1] Some material from this chapter are taken from the following papers:

1. M. O. Faruque, V. Dinavahi, "Real-Time HIL Simulation of VSC-HVDC and Experimental Validation", *submitted to IEEE Trans. on Power Delivery, December 2007.*
2. L-F. Pak, M. O. Faruque, X. Nie and V. Dinavahi, "A Versatile Cluster-Based Real-Time Digital Simulator for Power Engineering Research", *IEEE Trans. on Power Systems*, vol. 21, no. 2, pp. 455-465, May 2006.

A 6kW VSC-HVDC system experimental setup is used for validation of the real-time simulation results. In summary, this chapter focuses on:

- The modeling of VSC-HVDC system for off-line and real-time HIL simulation.
- The off-line simulation (using C-language) of the VSC-HVDC system.
- Use of discretization techniques such as TSSIT and RIT (proposed in Chapter 2) in both off-line and HIL simulation.
- Application of switching synchronization algorithms described in Chapter 3 for switching delay correction.
- Introduction of RTX-LAB real-time simulator and its use in both HIL simulation and experimental setup.
- Description of the 6kW experimental system.
- Comparison of off-line, real-time HIL-simulation and experimental results.

5.2 Modeling of VSC-HVDC for Real-Time Simulation

The equivalent circuit diagram of the VSC-HVDC system is shown in Fig. 5.1. The VSC 1 and VSC 2 are the two 6-pulse converters that operate on PWM control and establish the link between the DC side and the AC sides of the system. The modeling of different parts of the VSC-HVDC system for real-time simulation is given in the following sections.

5.2.1 AC System

The AC systems on both sides of the HVDC network are modeled as pure sinusoidal voltages:

$$v_{1a} = v_{2a} = V_{pk}\cos(\omega t) \tag{5.1}$$

$$v_{1b} = v_{2b} = V_{pk}\cos(\omega t - 2\pi/3) \tag{5.2}$$

$$v_{1c} = v_{2c} = V_{pk}\cos(\omega t + 2\pi/3) \tag{5.3}$$

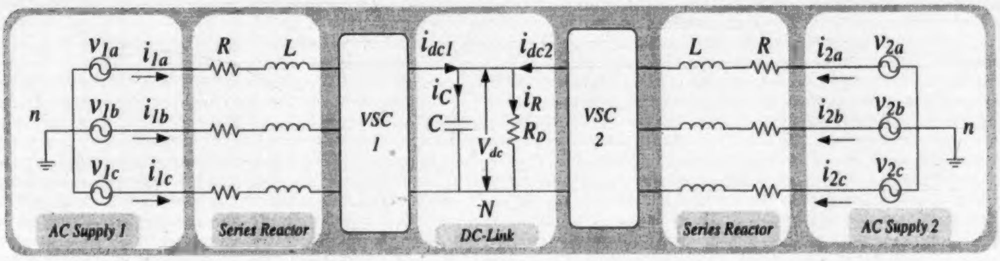


Figure 5.1: Equivalent circuit diagram of the VSC-HVDC test system.

where, V_{pk} is the peak supply voltage. In this case, V_{pk} is taken as $165.48V$ as this is the peak voltage used in the experiment setup in the RTX-Lab. The same supply is used for both sides of the system for two main reasons: 1) it simplifies the control algorithm and 2) it maintains the similarity between experiment and simulation supplies. A $2.5mH$ inductance is modeled as a series reactor with a resistance of 0.2Ω. The resistance, $R = 0.2\Omega$ accounts for the resistance of the inductor, resistance of the wire/cable connecting them in the circuit and others. Inductors and resistances are considered same ($R_a = R_b = R_c = R$ and $L_a = L_b = L_c = L$) in all three phases even though a ±5% mismatch has been observed in their values when measured with a $60Hz$ supply.

5.2.2 Converters

For modeling the converters, the switching function model is used where the converters are replaced by a circuit consisting of controlled voltage and current sources. The equivalent model of the voltage source converter is shown in Fig. 5.2. The relationship between the input and output quantities for converters with voltage-type input can be expressed as,

$$v_o(t) = f(t) \cdot v_{in}(t) \tag{5.4}$$

$$i_{in}(t) = g(t) \cdot i_o(t) \tag{5.5}$$

In the above equations, $f(t)$ and $g(t)$ are the switching functions. Usually, switching functions can be obtained by directly inspecting the converter circuit. Fig.5.2(b) shows three-phase voltage source converters represented through switching functions.

S_1, S_3 and S_5 are the pulse width modulation (PWM) gating functions for switches 1, 3, and 5, respectively. They have two levels (1 and 0). To get the instantaneous voltages showing high frequency switchings, voltages can be expressed in terms of the discrete switching functions S_k, $k = \{1, 3, 5\}$ which control the upper switches in each leg of the converter. When $S_k = 1$, the switch is on and when $S_k = 0$, the switch is off. The lower switches in each leg of the converter are switched in a complementary manner. The output voltages of the converter with respect to the negative DC bus N are given as

$$v_{kN} = S_k\, V_{dc}\ ;\ k = \{1, 3, 5\} \tag{5.6}$$

Under balanced conditions, the converter output voltages with respect to the AC system neutral n are given as

$$v_{an} = \frac{2}{3}v_{aN} - \frac{1}{3}(v_{bN} + v_{cN}) \tag{5.7}$$

$$v_{bn} = \frac{2}{3}v_{bN} - \frac{1}{3}(v_{cN} + v_{aN}) \tag{5.8}$$

$$v_{cn} = \frac{2}{3}v_{cN} - \frac{1}{3}(v_{aN} + v_{bN}) \tag{5.9}$$

or,

$$v_{kn} = \frac{2}{3}v_{kN} - \frac{1}{3}\sum_{\substack{i=\{a,b,c\}\\ i\neq k}} v_{iN} \tag{5.10}$$

The simulation is faster using the switching function models with almost the same results as the ideal-model approach gives. However, since individual switches no longer exist, it is not possible to monitor the voltage/current or conduction of individual switches. Neglecting high frequency components and the dc-offsets in the converter voltage, the relationship between V_{dc} and the average inverter phase voltages, v_a, v_b and v_c can be approximated through the following equations:

$$v_a = \frac{m_a V_{dc}}{2}\cos(\omega t - \delta) \tag{5.11}$$

$$v_b = \frac{m_a V_{dc}}{2}\cos(\omega t - \delta - 2\pi/3) \tag{5.12}$$

$$v_b = \frac{m_a V_{dc}}{2}\cos(\omega t - \delta + 2\pi/3) \tag{5.13}$$

where, m_a and δ are modulation index and phase shift, respectively, obtained from the

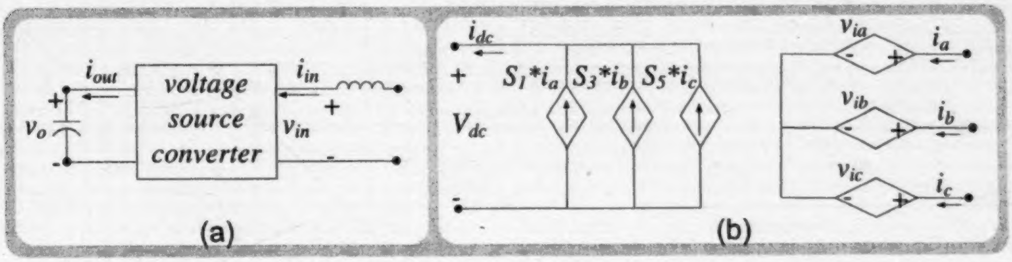


Figure 5.2: Switching function model of voltage source converter.

controller. This relationship, however, gives only the average voltage at the inverter terminal.

5.2.3 DC-Link

The DC-link in the system consists of a DC capacitor, C and a discharging resistor, R_D. The voltage in the DC-link is maintained by controller 1 which uses PWM to keep the DC-link voltage constant. The voltage and currents in the DC-link are given as:

$$\begin{aligned} i_{dc1} &= S_{11}i_{1a} + S_{13}i_{1b} + S_{15}i_{1c} && (5.14) \\ i_{dc2} &= S_{21}i_{2a} + S_{23}i_{2b} + S_{25}i_{2c} && (5.15) \\ C\frac{dV_{dc}}{dt} &= i_{dc1} + i_{dc2} - \frac{V_{dc}}{R_D} && (5.16) \end{aligned}$$

where, S_{11}, S_{13}, S_{15} are the states (1 or 0) of the top IGBTs of VSC 1 and S_{21}, S_{23}, S_{25} are the states of the top IGBTs of VSC 2.

5.2.4 Controller

The control scheme described in Chapter 4 is implemented in the following three cases: (1) off-line simulation using C code, (2) real-time HIL simulation and 3) experimental validation. From (4.15), the decoupled d-q control technique yields,

$$\begin{aligned} v_{1id} &= L(\omega i_{1q} - x_1) + |v_1| \\ v_{1iq} &= L(-\omega i_{1d} - x_2) && (5.17) \end{aligned}$$

for VSC 1 and for VSC 2, the equations are,

$$\begin{aligned} v_{2id} &= L(\omega i_{2q} - x_1) + |v_2| \\ v_{2iq} &= L(-\omega i_{2d} - x_2) \qquad (5.18) \end{aligned}$$

The modulation index m_{a1} and m_{a2} and the phase shift angle, δ_1 and δ_2 are calculated using the following equations:

$$\begin{aligned} m_{a1} &= \frac{2\left(\sqrt{v_{1id}^2 + v_{1iq}^2}\right)}{V_{dc}} \qquad (5.19) \\ \delta_1 &= tan^{-1}\left(\frac{v_{1iq}}{v_{1id}}\right) \qquad (5.20) \\ m_{a2} &= \frac{2\left(\sqrt{v_{2id}^2 + v_{2iq}^2}\right)}{V_{dc}} \qquad (5.21) \\ \delta_2 &= tan^{-1}\left(\frac{v_{2iq}}{v_{2id}}\right) \qquad (5.22) \end{aligned}$$

The three-phase feference modulating signals are created from these modulation indices, m_{a1} and m_{a2}, and the phase shift angles, δ_1 and δ_2. The reference signals are then compared with the triangular carrier wave to produce the required gate pulses. The switching functions of the converter, S_k changes its values (1 or 0), based on the value of the gating pulses. If the gate is high, $S_k = 1$, else $S_k = 0$.

5.2.5 State-Space Model of the Overall System

The complete VSC-HVDC system consists of two ac supplies, two phase reactors, two converters, DC-link and a digital controller. The modeling of the entire system has been carried out in light of the aformentioned modeling technique of various parts of the network. The differential equations for converter 1 side of the VSC-HVDC system can be written as

$$\begin{aligned} L\frac{di_{1a}}{dt} &= -Ri_{1a} - v_{1ia} + v_{1a} \qquad (5.23) \\ L\frac{di_{1b}}{dt} &= -Ri_{1b} - v_{1ib} + v_{1b} \qquad (5.24) \\ L\frac{di_{1c}}{dt} &= -Ri_{1c} - v_{1ic} + v_{1c} \qquad (5.25) \end{aligned}$$

where, v_{1a}, v_{1b}, v_{1c} are the supply voltage of AC side 1 and $v_{1ia}, v_{1ib}, v_{1ic}$ are the voltage at the terminal of VSC 1. Similarly, for VSC 2, with v_{2a}, v_{2b}, v_{2c} as the supply voltage of AC side 2, and $v_{2ia}, v_{2ib}, v_{2ic}$ as the voltage at the terminal of VSC 2, the equations are:

$$\begin{aligned} L\frac{di_{2a}}{dt} &= -Ri_{2a} - v_{2ia} + v_{2a} && (5.26) \\ L\frac{di_{2b}}{dt} &= -Ri_{2b} - v_{2ib} + v_{2b} && (5.27) \\ L\frac{di_{2c}}{dt} &= -Ri_{2c} - v_{2ic} + v_{2c} && (5.28) \end{aligned}$$

The DC-link voltage can be expressed as

$$\begin{aligned} \frac{dV_{dc}}{dt} &= S_{11}i_{1a} + S_{13}i_{1b} + S_{15}i_{1c} && (5.29) \\ &+ S_{21}i_{2a} + S_{23}i_{2b} + S_{25}i_{2c} - \frac{V_{dc}}{CR_D} \end{aligned}$$

Combining (5.23), (5.26) and (5.29), the state-space equation can be formed and written as:

$$\dot{\mathbf{x}}(t) = \mathbf{A_c}\mathbf{x}(t) + \mathbf{B_c}\mathbf{u}(t) \qquad (5.30)$$

where, the state matrix, $\mathbf{x} = \begin{bmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \\ i_{2a} \\ i_{2b} \\ i_{2c} \\ V_{dc} \end{bmatrix}$ and the inputs are $\mathbf{u} = \begin{bmatrix} v_{1a} - v_{1ia} \\ v_{1b} - v_{1ib} \\ v_{1c} - v_{1ic} \\ v_{2a} - v_{2ia} \\ v_{2b} - v_{2ib} \\ v_{2c} - v_{2ic} \\ 0 \end{bmatrix}$. The system matrices are

$$\mathbf{A_c} = \begin{bmatrix} -\frac{R}{L} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{R}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{R}{L} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{R}{L} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{R}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{R}{L} & 0 \\ S_{11} & S_{13} & S_{15} & S_{21} & S_{23} & S_{25} & -\frac{1}{CR_D} \end{bmatrix}$$

$$\mathbf{B}_c = \begin{bmatrix} \frac{1}{L} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{L} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{L} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{L} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

5.2.6 Discrete-Time Model of the Overall System

Any digital simulation using a computer needs discretization of its systems equations irrespective of whether nodal method or state-space method is used for solving the system. The discretization of system matrices can be done using any method such as SIT, TSSIT or RIT and state-space solution can be performed. Also individual elements can be discretized using Trapezoidal Rule, SIT/TSSIT or RIT and can be solved using nodal analysis. Discretization of the above system equations using Trapezoidal Rule with Δt as the time-step will give the following equations:

$$\begin{aligned} i_{1a}(t) &= A[i_{1a}(t-\Delta t)] + B[v_{1a}(t) + v_{1a}(t-\Delta t)] && (5.31) \\ &- B[v_{1ai}(t) + v_{1ai}(t-\Delta t)] \\ i_{1b}(t) &= A[i_{1b}(t-\Delta t)] + B[v_{1b}(t) + v_{1b}(t-\Delta t)] && (5.32) \\ &- B[v_{1bi}(t) + v_{1bi}(t-\Delta t)] \\ i_{1c}(t) &= A[i_{1c}(t-\Delta t)] + B[v_{1c}(t) + v_{1c}(t-\Delta t)] && (5.33) \\ &- B[v_{1ci}(t) + v_{1ci}(t-\Delta t)] \end{aligned}$$

and,

$$\begin{aligned} i_{2a}(t) &= A[i_{2a}(t-\Delta t)] + B[v_{2a}(t) + v_{2a}(t-\Delta t)] && (5.34) \\ &- B[v_{2ai}(t) + v_{2ai}(t-\Delta t)] \\ i_{2b}(t) &= A[i_{2b}(t-\Delta t)] + B[v_{2b}(t) + v_{2b}(t-\Delta t)] && (5.35) \\ &- B[v_{2bi}(t) + v_{2bi}(t-\Delta t)] \\ i_{2c}(t) &= A[i_{2c}(t-\Delta t)] + B[v_{2c}(t) + v_{2c}(t-\Delta t)] && (5.36) \\ &- B[v_{2ci}(t) + v_{2ci}(t-\Delta t)] \end{aligned}$$

where, $A = \frac{1-\frac{\Delta t}{2}\frac{R}{L}}{1+\frac{\Delta t}{2}\frac{R}{L}}$, $B = \frac{\frac{\Delta t}{2L}}{1+\frac{\Delta t}{2}\frac{R}{L}}$, $v_{1k}(k = a, b, c)$ and $v_{2k}(k = a, b, c)$ are the supply voltages given in (5.1), (5.4) and (5.6), and $v_{1ki}(k = a, b, c)$ and $v_{2ki}(k = a, b, c)$ are the converter voltages given in (5.10).

If TSSIT is used instead of Trapezoidal Rule, the currents will be given by the following equations:

$$i_{1a}(t) = E[i_{1a}(t - \Delta t)] + F[v_{1a}(t) - v_{1ai}(t)] \tag{5.37}$$

$$i_{1b}(t) = E[i_{1b}(t - \Delta t)] + F[v_{1b}(t) - v_{1bi}(t)] \tag{5.38}$$

$$i_{1c}(t) = E[i_{1c}(t - \Delta t)] + F[v_{1c}(t) - v_{1ci}(t)] \tag{5.39}$$

and

$$i_{2a}(t) = E[i_{2a}(t - \Delta t)] + F[v_{2a}(t) - v_{2ai}(t)] \tag{5.40}$$

$$i_{2b}(t) = E[i_{2b}(t - \Delta t)] + F[v_{2b}(t) - v_{2bi}(t)] \tag{5.41}$$

$$i_{2c}(t) = E[i_{2c}(t - \Delta t)] + F[v_{2c}(t) - v_{2ci}(t)] \tag{5.42}$$

where, $E = e^{-\frac{R\Delta t}{L}}$ and $F = \frac{1}{R}(1 - e^{-\frac{R\Delta t}{L}})$.

For RIT, the equations will be given as follows:

$$\begin{aligned} i_{1a}(t) &= E[i_{1a}(t - \Delta t)] + G[v_{1a}(t) - v_{1ai}(t)] \\ &+ H[v_{1a}(t - \Delta t) - v_{1ai}(t - \Delta t)] \end{aligned} \tag{5.43}$$

$$\begin{aligned} i_{1b}(t) &= E[i_{1b}(t - \Delta t)] + G[v_{1b}(t) - v_{1bi}(t)] \\ &+ H[v_{1b}(t - \Delta t) - v_{1bi}(t - \Delta t)] \end{aligned} \tag{5.44}$$

$$\begin{aligned} i_{1c}(t) &= E[i_{1c}(t - \Delta t)] + G[v_{1c}(t) - v_{1ci}(t)] \\ &+ H[v_{1c}(t - \Delta t) - v_{1ci}(t - \Delta t)] \end{aligned} \tag{5.45}$$

and

$$\begin{aligned} i_{2a}(t) &= E[i_{2a}(t - \Delta t)] + G[v_{2a}(t) - v_{2ai}(t)] \\ &+ H[v_{2a}(t - \Delta t) - v_{2ai}(t - \Delta t)] \end{aligned} \tag{5.46}$$

$$\begin{aligned} i_{2b}(t) &= E[i_{2b}(t - \Delta t)] + G[v_{2b}(t) - v_{2bi}(t)] \\ &+ H[v_{2b}(t - \Delta t) - v_{2bi}(t - \Delta t)] \end{aligned} \tag{5.47}$$

$$\begin{aligned} i_{2c}(t) &= E[i_{2c}(t - \Delta t)] + G[v_{2c}(t) - v_{2ci}(t)] \\ &+ H[v_{2c}(t - \Delta t) - v_{2ci}(t - \Delta t)] \end{aligned} \tag{5.48}$$

where, $G = \frac{R\Delta t - L + Le^{-\frac{R\Delta t}{L}}}{R^2\Delta t}$ and $H = \frac{L - Le^{-\frac{R\Delta t}{L}} - R\Delta t e^{-\frac{R\Delta t}{L}}}{R^2\Delta t}$. In these equations, A, B, E, F, G and H are constants whose values depend on the system parameters and the time-step Δt. These constants remain same unless system parameters or time-step is changed and are calculated once at the beginning of the simulation. The DC side equations are

$$\begin{aligned}
i_{dc1}(t) &= S_{11}(t)i_{1a}(t) + S_{13}(t)i_{1b}(t) + S_{15}(t)i_{1c}(t) && (5.49)\\
i_{dc2}(t) &= S_{21}(t)i_{2a}(t) + S_{23}(t)i_{2b}(t) + S_{25}(t)i_{2c}(t) && (5.50)\\
V_{dc}(t) &= I[i_{dc1}(t) + i_{dc1}(t-\Delta t) + i_{dc2}(t) + i_{dc2}(t-\Delta t)] && (5.51)\\
&+ J[V_{dc}(t-\Delta t)]
\end{aligned}$$

I and J are again two constants whose values depend on the parameters, time-step and also on the method of discretization used for simulation.

5.3 Off-line Simulation using C

In the off-line simulation, the entire VSC-HVDC system and its controllers are coded using C language. The discretization techniques described in Chapter 2 and switching delay correction algorithms described in Chapter 3 were applied in the C-program. Nodal analysis is used with the elements discretized using RIT for steady-state and using TSSIT at some transient situations. The objective of performing another off-line simulation using C program is to ensure that the modeling and solution technique described in Section 5.2 and 5.3 are accurate, efficient and would have no difficulty in the implementation of real-time simulation. The same C-program with some modifications (to make it compatible with the s-function template) is used for real-time HIL simulation. In both cases, the system is modeled and solved using the equations described in Section 5. The control technique described in Chapter 4 is implemented. The simulation is carried out using Microsoft Visual C/C++ v.6.0. A $2kHz$ switching frequency is used for the carrier and $20\mu s$ simulation time-step is used for the off-line simulation. As the control signals are very low frequency ($60Hz$) signals, it is redundant to sample them at $20\mu s$ time-step. Therefore, multi-rate sampling has been used where the control is sampled at lower frequency ($100\mu s$, five times of the simulation time-step). The results from this off-line simulation will be presented in Section 5.8 along with the HIL-simulation and experimental results.

5.4 Real-Time Digital Simulator

The study of HIL simulation of the VSC-HVDC was conducted in the RTX-Lab at the University of Alberta. The lab is equipped with a powerful, state-of-the-art real-time simulator as shown in Fig. 5.3 which is built to perform large computational tasks. The simulator is built out of Commercial-Of-The-Shelf (COTS) components such as general purpose processor based high speed computers as the main computation engine. Standard computers are used for model development, and FPGA based I/Os are used for communication with the external hardware. InfiniBand and Gigabit ethernet network are used for communication between various computers. Eight dual Intel Xeon processor based 3.0 GHz Hyper-threaded (HT) computer ($8 \times 2 = 16$ processors) known as *Targets* or *Target Nodes* are used in parallel for carrying out all the real-time computation of large systems. There are also standard 3.0 GHz Pentium IV processor based computers known as *Hosts* or *Command Stations* which are used for model preparation and monitoring of the simulation results. As shown in the hardware architecture of the simulator in Fig. 5.4, the *Target* computer facilitates the connection of external hardware through the FPGA-based I/O ports and a data transfer link between the *Host* and the *Target* through a Gigabit ethernet link. Signal-wire and InfiniBand are two other fast communication links used for data transfer between nodes when a model is split into multiple nodes. Real-time simulation results can be observed through an oscilloscope or other hardware devices connected to the I/O terminals of the *Target* node.

The *Target* runs on a Linux based real-time operating system which offers optimized performance through a single programming environment and direct control of all system operations by using single kernel design. It offers eXtra High Performance (XHP) mode operation through CPU shielding where one CPU is dedicated for the simulation while the other CPU is responsible for other jobs such as interrupt handling, writing to the disk and I/O operations. The *Target* is responsible for real-

Figure 5.3: Snapshot of the PC-Cluster based real-time digital simulator installed in the RTX-Lab.

time execution of the model, data transfer between nodes and the *Host*, and data communication with external hardware through I/Os. The I/Os operate through an FPGA based signal conditioning board (OP5110) which is inserted in the PCI-X bus of the *Target* node. The A/D and D/A cards are connected at the end of the FPGA card to facilitate data sampling from or to the real world.

The *Host* computer runs on Windows XP on which a real-time interfacing software RT-LAB [15] from Opal RT Technologies Inc. is installed to perform the co-ordination of all the hardware involved in the simulation. The *Host* also runs other programs and GUIs to prepare, edit and customize the simulation model. MATLAB/SIMULINK and the SimPowerSystems toolbox are the primary tools used to prepare the model

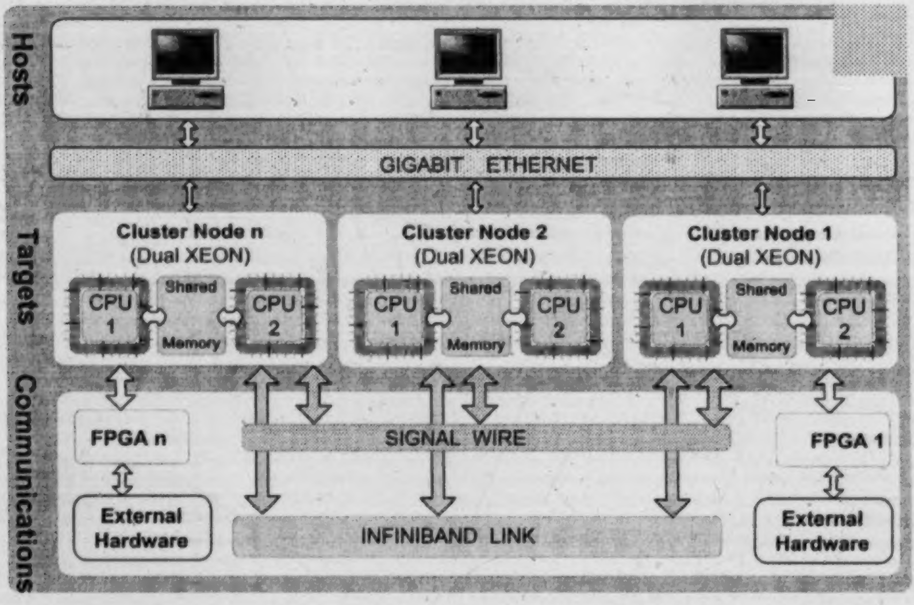


Figure 5.4: Hardware architecture of the simulator installed in the RTX-Lab.

for simulation. The detailed description of the simulator and its operation is available in [13].

5.5 HIL Simulation of VSC-HVDC

For HIL simulation of the VSC-HVDC system, the electrical circuit is modeled, coded in C, wrapped up in the s-function format and then simulated through a SIMULINK model. The model runs inside a *Target* node of the RTX-Lab simulator. A digital controller is used to generate the gate pulses which are sent to the simulator through the necessary I/O interfaces. The digital controller receives feedback signals from the simulator, processes them in the hardware, generates the $15V$ gate signals and then send them to the simulator. A separate *Target* node is used as hardware for the digital controller with one additional FPGA card inserted in that node. The new FPGA carrier board (OP5130) is dedicated to generate PWM gating signals while the built-in FPGA (OP5110) controls the I/O operations and perform other functions such as simulation synchronization, signal conditioning, etc. Fig. 5.5 shows

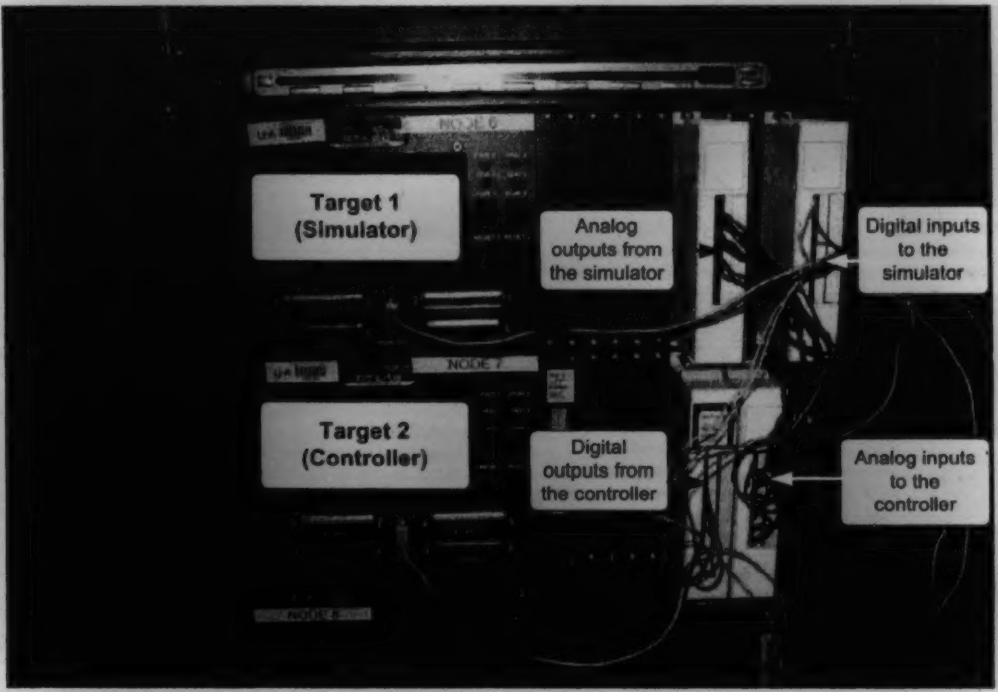

NODE 6
Target 1
(Simulator)
Analog outputs from the simulator
Digital inputs to the simulator
NODE 7
Target 2
(Controller)
Digital outputs from the controller
Analog inputs to the controller
NODE 8

Figure 5.5: Snapshot of the hardware setup for HIL simulation of the VSC-HVDC.

the photograph of HIL simulation setup while the next section gives details about the setup.

5.5.1 Hardware Setup

The hardware setup for HIL real-time simulation is shown in Fig. 5.6. *Target* 1 is used as the simulator while *Target* 2 is used as the controller. *Host* 1 is used to prepare the model for simulation. It can also be used to monitor the real-time simulation results collected from *Target* 1. Similarly, *Host* 2 is used for developing the control scheme and once the controller is ready, it will be loaded on *Target* 2 for implementation. Therefore, *Target* 2 works as the digital controller whose inputs are analog feedback signals ($\pm 15V$) and the outputs are gate signals ($0 - 15V$) that are sent to the simulator for simulating the system. The HIL simulation and the controller are independently controlled by both *Host* 1 and *Host* 2.

Target 1, the simulator has one FPGA but *Target* 2, the controller has two FPGAs.

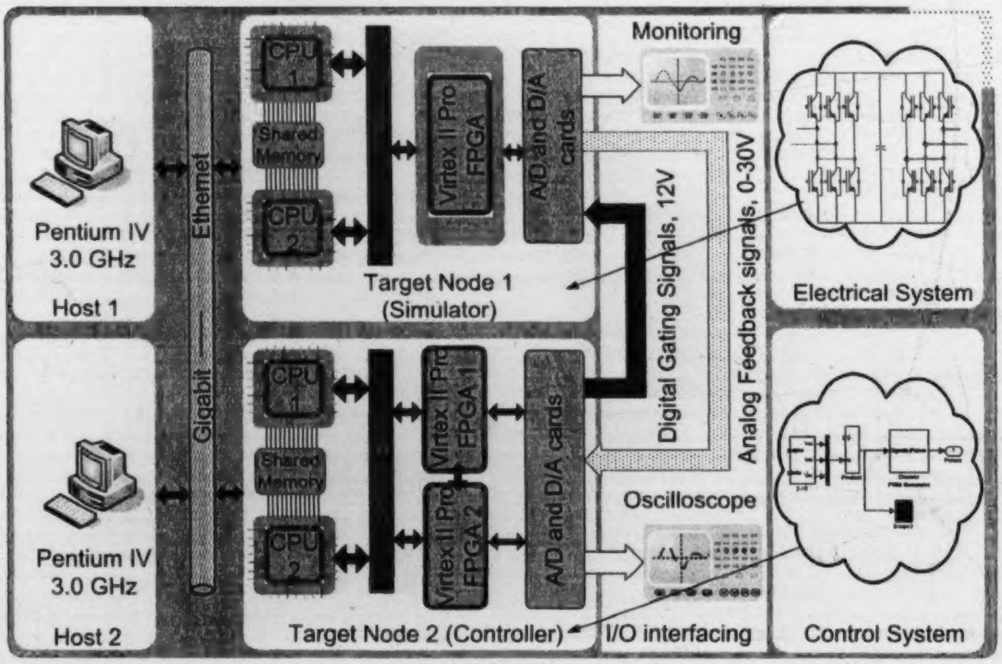


Figure 5.6: Block layout of the hardware setup for HIL simulation of the VSC-HVDC.

The additional FPGA (OP5130) in *Target* 2 is dedicated for generating gating signals by comparing the carrier signals (generated inside the FPGA) with the reference signals (produced by the control algorithm in the Xeon CPU). The FPGA on the OP5130 carrier board is a Xilinx Virtex-II Pro that has 11,088 logic cells and an IBM power processor in it, and it operates at a frequency of $100MHz$. This gives a $10ns$ resolution which is the time-step used for generating triangular carrier signal for PWM. On one side, the FPGA OP5110 is connected to the PCI-X slot of the computer and on the other side it could be connected to the FPGA OP5130 board or to the I/O carriers so that signals can be transmitted to and from the external device to the Xeon CPU. Fig. 5.7 shows the OP5110 FPGA board used in *Target* 1 and *Target* 2, and Fig. 5.8 shows the OP5130 FPGA board used in *Target* 2 for HIL simulation.

Each I/O carrier has two boards in it; A/D (Analog-to-Digital) card and D/A (Digital-to-Analog) card. The A/D card has 16 channels and each channel converts

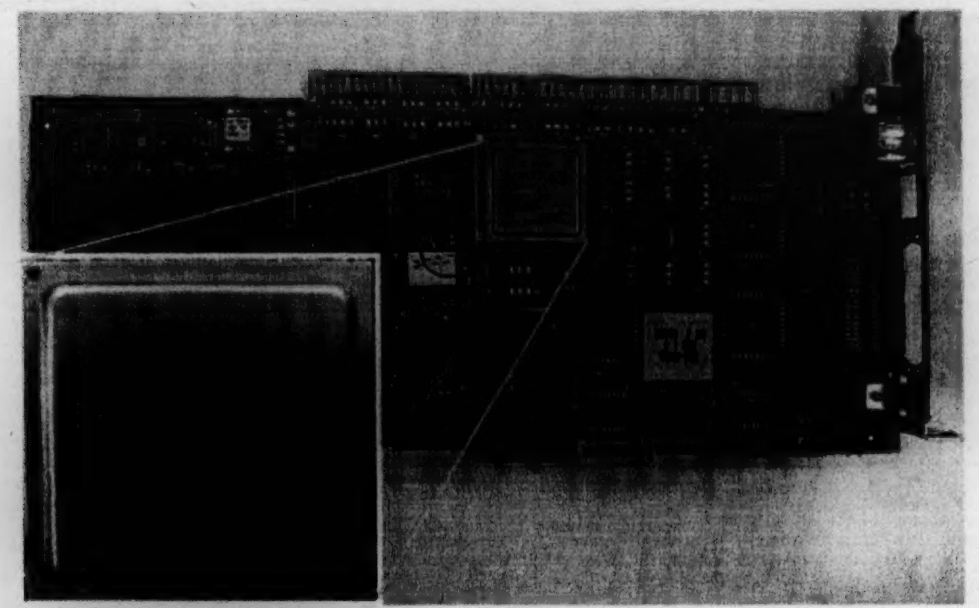


Figure 5.7: OP5110 FPGA module in the PCIX slot of Target node.

an analog voltage signal into a 16-bit digital signal. The default range of analog signal is $\pm 15V$, however, this can be increased to a maximum of $\pm 30V$ by changing the internal resistance of the board. Similarly, the D/A board receives 16-bit digital signals and produces an analog voltage in the range of $\pm 15V$. The sampling frequency of the boards are $1MHz$ and a time-delay of $80ns$ is taken by the conversion process. The 16-channel digital output signals are generated in the range of $0 - 5V$ from the FPGA which is further amplified to the range of $0 - 15V$. The digital signals at a higher voltage level are required to drive IGBT based converters through the coupling of gating circuitry. Opto-couplers are used to isolate the I/O boards from external hardware so that any damage to the simulator can be avoided.

5.6 Implementation of the HIL Simulation

SIMULINK, RTLAB, SimPowerSystems and C program are the main tools used for developing, loading and running the models for HIL simulation. The *Host* 1 is used for developing the system model and the *Host* 2 is used for developing the control

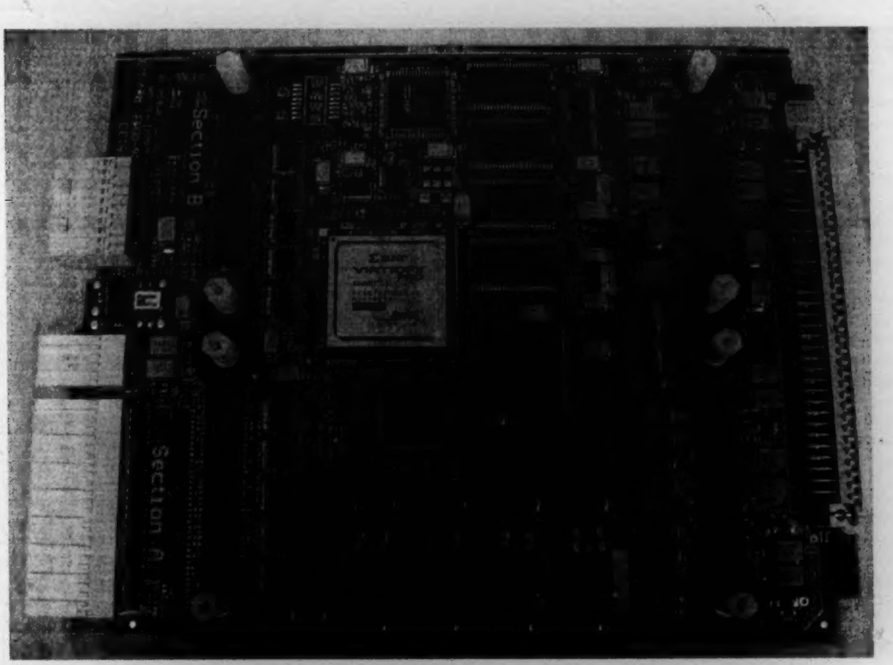


Figure 5.8: OP5130 FPGA module in the carrier slot of Target node.

algorithm. For modeling the electrical system of VSC-HVDC, s-function is used to wrap the C-program in the SIMULINK environment. Once the models were prepared, they were loaded in two *Target* nodes using RTLAB. As shown in Fig. 5.5, the two *Targets* are physically connected through I/O channels using standard wires. The I/O channels are used to transmit analog and digital signals between *Targets*. The number of I/O channels used depends on the system and its controller implementation. As shown in Fig. 5.9, in this simulation, six $0 - 15V$ gating signals (digital signals) come out through the I/O of *Target 2*. From *Target* 2, they are fed to the simulator through the I/O of *Target* 1. The simulation program (C-program) in *Target* 1 receives these gating signals and perform the transient simulation of electrical system and produces the required results. The results can either be stored or monitored through external devices such as an oscilloscope or the *Host* computer. Even though there are twelve IGBTS in the system, the reason for sending six gate pulses is due to the complimentary nature of the gate pulses. The six pulses belong to the top IGBT of each leg and the gate pulses for the bottom IGBT are the compliments of

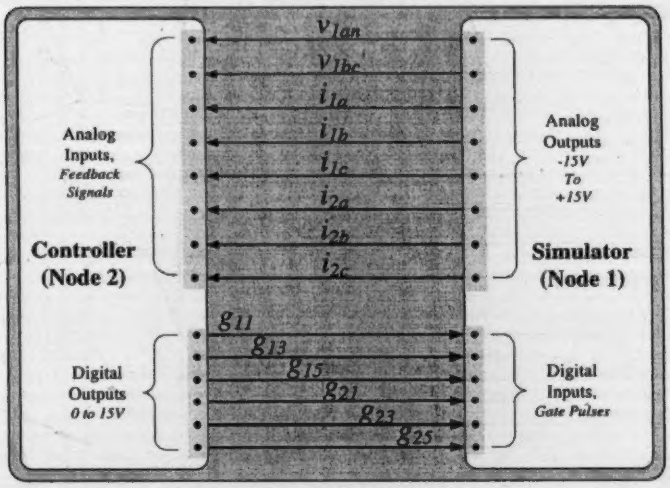


Figure 5.9: I/O connections between the simulator and the controller.

the gate pulses of the top IGBTs. *Target* 1 also sends the necessary feedback signals, mostly analog signals to *Target* 2 for implementing the control algorithm and thus to produce the gating signals for the next time-step. In this simulation, minimum eight (8) analog signals ($v_{1an}, v_{1bc}, i_{1a}, i_{1b}, i_{1c}, i_{2a}, i_{2b}$ and i_{2c}) are necessary for the control algorithm to function properly. v_{1an} and v_{1bc} are used to generate the Phase-Lock-Loop (PLL) signals and the currents are used for the control algorithm.

5.6.1 Digital Controller

The control algorithm described in Chapter 4 is implemented in SIMULINK. Standard blocks from SimPowerSystems and Control toolbox are used to construct the model. The reason for using the SIMULINK platform for controller implementation is its ability to interface with other programs such as RT-LAB, which is used to load the model in the *Target* node. The scope of using a *Target* node as a digital controller is one of the many unique features of the RTX-Lab simulator. In case of unavailability of a standard function to perform any particular job, a math function is used where an arbitrary mathematical formula can be inserted inside the function. The control algorithm is implemented inside the processors of *Target* 2, however, the PWM is generated inside the Xilinx FPGA or the FPGA 2 in OP5130 carrier board as shown in Fig. 5.6.

As shown in Fig. 5.9, the digital controller receives analog signals, $v_{1an}, v_{1bc}, i_{1a}, i_{1b}, i_{1c}, i_{2a}, i_{2b}$ and i_{2c} from the simulator and converts them into digital signals with the help of A/D card inserted in *Target 2*. The digitized signals are then passed to the Xeon processors through the FPGA on the OP5110 board. The SIMULINK model for the control algorithm receives those signals, perform $abc - \alpha\beta - dq$ conversion. The feedback currents are two sets of signals and would be used for two controllers for the two converters (VSC 1 and VSC 2) in the VSC-HVDC system. The equations described in Section 4 of Chapter 4 are implemented to generate modulation index, m_{a1}, m_{a2}, δ_1 and δ_2 for both controller 1 and controller 2. Using these m_a and δ, two sets of three-phase sinusoidal modulating reference signals are generated for both controllers using the following equations:

$$v_{1aref} = m_{a1}\sin(\omega t + \delta_1) \tag{5.52}$$

$$v_{1bref} = m_{a1}\sin(\omega t + \delta_1 - 2\pi/3) \tag{5.53}$$

$$v_{1cref} = m_{a1}\sin(\omega t + \delta_1 + 2\pi/3) \tag{5.54}$$

and,

$$v_{2aref} = m_{a2}\sin(\omega t + \delta_2) \tag{5.55}$$

$$v_{2bref} = m_{a2}\sin(\omega t + \delta_2 - 2\pi/3) \tag{5.56}$$

$$v_{2cref} = m_{a2}\sin(\omega t + \delta_2 + 2\pi/3) \tag{5.57}$$

It is to be noted that the amplitude of the reference sinusoidal signals are limited in the range of $\pm 1V$. This imposes a restriction on the value of m_{a1} and m_{a2} to be in the linear range of $0 - 1$. A limiter is used to force them within that range if their values exceed the range.

Next, a data type conversion is performed where, the reference signals are converted to 16-bit digital format from *double* format. These signals are then passed to the OP5130 FPGA, where PWM gate pulses are generated. The developed SIMULINK controller model is shown in Appendix-C.

The control scheme uses cascaded PI controllers for controlling v_{dc} and i_d in the controller 1 where i_d is controlled through a inner loop feedback control and v_{dc} is controlled through a outer-loop feedback control. Of these, the inner loop PI response needs to be faster than that of the outer loop in order for the cascaded combination to work. This is due to the faster response of the ac currents than that of the DC-link voltage. As the system parameters appear to be slightly different for HIL-simulation and experiment, a slight re-tuning is necessary for all the gains of the PI controllers.

5.6.2 PWM Generation Inside FPGA

The technique of generating PWM inside an FPGA has been described in [56]. To generate PWM gate signals, two types of signals are necessary: (1) the three-phase reference signals at $60Hz$ and (2) a high frequency triangular carrier signal. The three-phase reference signals are generated by the control algorithm and then sent to the OP5130 FPGA, however, the carrier signal is generated inside the FPGA at a fine resolution. The frequency of the carrier signal is chosen as $2kHz$ but the reference signals are sampled at $4kHz$. A separate synchronizing pulse is also fed to the FPGA which has a frequency of $2kHz$ and a duty ratio of 50%. This is used to synchronize the reference signals with the FPGA counter that generates the carrier signal.

For both HIL simulation and experimental verification, exactly the same control technique and the same PWM generation method is used. As shown in Fig. 5.10, a $2kHz$ carrier signal gives a period of $500\mu s$, therefore, the carrier signal is generated with a period of $500\mu s$ using a 16-bit counter. The counter should reach to the top of the carrier leg ($+V_{tri}$) from the bottom of that leg ($-V_{tri}$) in $250\mu s$ and in another $250\mu s$, it should go back to the bottom of the leg. The FPGA has a clock period of $10ns$ (100MHz) and in $250\mu s$, it reaches from -12500 to 12500. In the next $250\mu s$, the counter goes back to -12500 thus producing one cycle of the carrier wave. Because the counter varies between -12500 to 12500, the modulating signals (three-phase reference signals) should also be varied in the same range to maintain a linear modulation index, where 12500 would be the maximum amplitude of the sine-wave and -12500 would be the minimum amplitude. A comparator is used for each phase to produce high or low outputs which translate into the required gate pulses for the top IGBT of that

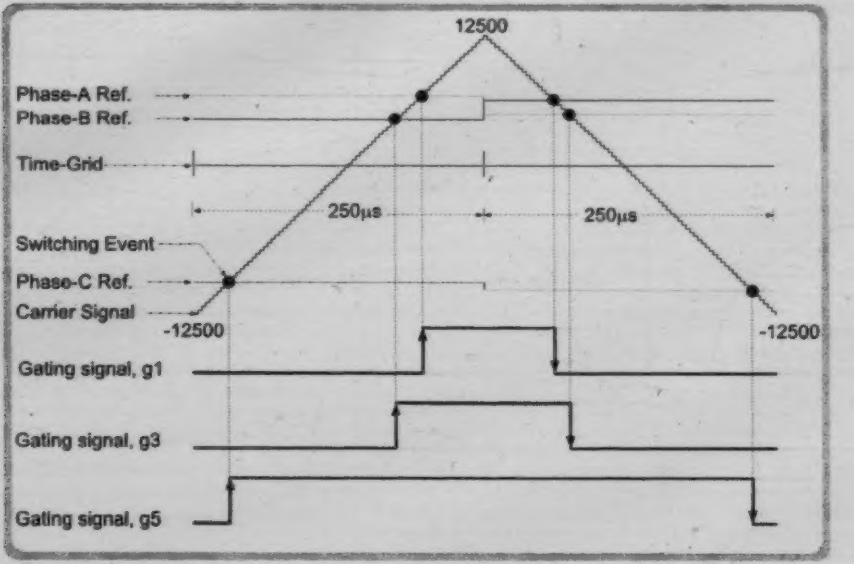


Figure 5.10: PWM Generation inside FPGA

particular phase. The gate pulse for the bottom IGBT is taken as the complementary of the top IGBT gate signal and implemented through a NOT gate. A logic block diagram can be used to implement the dead time by adding some delay for the gate pulse of the bottom IGBT. In this setup, no dead-time is implemented inside the FPGA, however, for the experimental setup, dead-time is a must and implemented in the gate drive board. The complete block diagram is implemented using Xilinx System Generator block-sets and are embedded in the SIMULINK model (shown in Appendix C). Fig. 5.11 shows the oscilloscope traces for the reference signal, the triangular carrier wave and the gate pulse generated for the top IGBT.

5.6.3 Modeling the Electrical System

The electrical system of the VSC-HVDC is modeled and solved using the equations described in Section 5.2.6. A C-program is developed and placed in the SIMULINK environment, however, it is not solved by any of the SIMULINK solvers. Instead, an arrangement is made to run the simulation in SIMULINK environment by using the solver specified inside the C-program. This is done with the help of a s-function

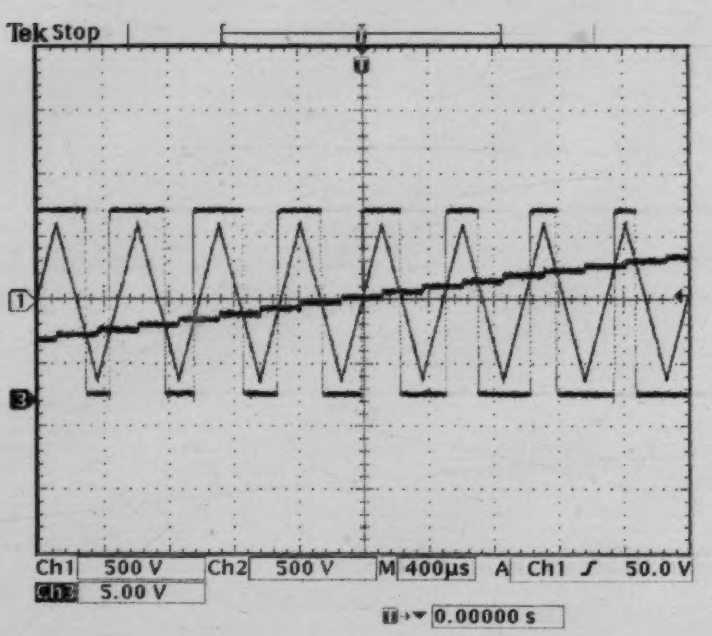
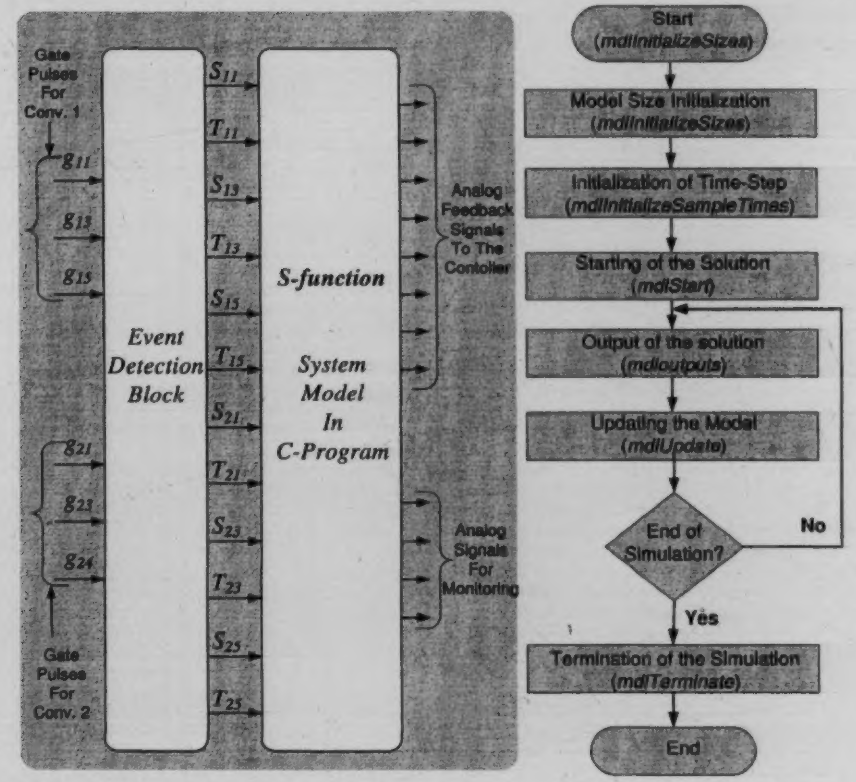


Figure 5.11: Oscilloscope traces for the PWM carrier, reference signal and gate pulse.

wrapper where the model and its solution is programmed in C-language and then embedded inside the s-function. The program is solved at every time-step assigned by the SIMULINK but the time-step can be changed from outside the s-function in SIMULINK main model. The change in time-step can be passed to the C-program by defining it as a parameter of the s-function. The '*discrete solver*' option is used for the simulation, even though selection of different solver in SIMULINK does not influence the s-function as the solver is chosen inside the C-program. It is possible to use multiple solvers inside the s-function if the C-program is designed accordingly.

5.6.4 Capturing of Switching Events

The inputs and outputs to the s-function are the outputs and inputs, respectively from the controller in *Target* 2. The s-function framework is shown in Fig. 5.12(a) where its inputs and outputs are also shown. The six gate pulses ($g_{ij}, i = 1,2$ and $j = 1,3,5$) are first sent to a *Event Detection Block* which determines the *States* ($S_{ij}, i = 1,2$, $j = 1,3,5$) of each gate pulse and their *Time Stamp* ($T_{ij}, i = 1,2,\ j = 1,3,5$). The


Gate Pulses For Conv. 1
g_{11}
g_{13}
g_{15}
g_{21}
g_{23}
g_{24}
Gate Pulses For Conv. 2
Event Detection Block
S_{11}
T_{11}
S_{13}
T_{13}
S_{15}
T_{15}
S_{21}
T_{21}
S_{23}
T_{23}
S_{25}
T_{25}
S-function
System Model In C-Program
Analog Feedback Signals To The Contoller
Analog Signals For Monitoring
Start (mdlInitializeSizes)
Model Size Initialization (mdlInitializeSizes)
Initialization of Time-Step (mdlInitializeSampleTimes)
Starting of the Solution (mdlStart)
Output of the solution (mdlOutputs)
Updating the Model (mdlUpdate)
End of Simulation?
No
Yes
Termination of the Simulation (mdlTerminate)
End

(a) (b)

Figure 5.12: The s-function (a) input and output terminals (b) flow chart.

Event Detection block is an RT-LAB block that comes with the RT-LAB software. This block is used to capture digital signals and to detect the transitions that occur on these signals during each calculation step. The outputs of the *Event Detector* block are the twelve signals of which six are the *States* of the gating signals and the rest six are their *Time-stamps.* In case of more than one switching in one time-step, both the *State*, S_{ij}, and its *Time-stamps*, T_{ij} are generated in the vector format. This means if two switching events happen in one time-step, the *State* outputs will be either [1 0] or [0 1] and the *Time-stamps* output will be $[T1T2]$, where $T1$ is the time of first switching and $T2$ is the time of second switching.

During each time-step, input lines are checked continuously for transitions.

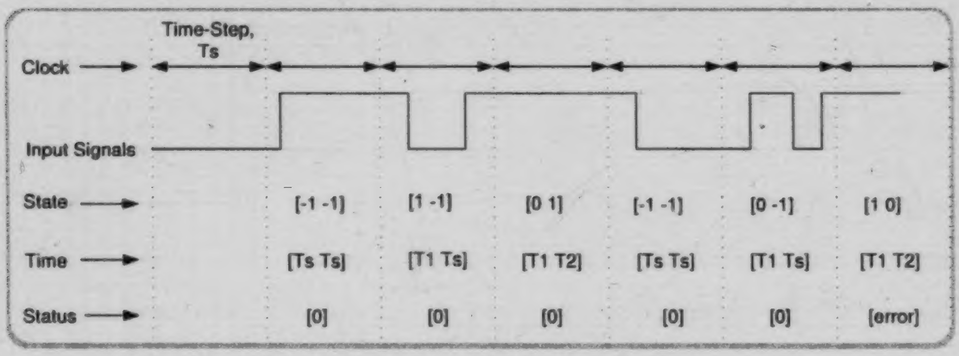


Figure 5.13: The truth table for event detection algorithm.

The hardware reports the moment of a *State* change relative to the start of the computation step. Rising, falling or both transitions can be reported as requested by the user. During one calculation step, all captured events are stored on the FPGA based event detector card and they are uploaded by the block at the beginning of the next calculation step. The *Event Detection* card can detect as many as 511 transitions per port, however, the number of events returned to the model is limited by the size of the output vectors defined by the user in the model. If the size of the *Time-stamps* and *States* vectors defined by the user is smaller than the actual number of events detected, the *Status* output pin returns an error signal. The truth table for *Event Detection* block is given in Fig. 5.13 where signal waveforms and returned *States* and *Time-stamps* vectors are shown. In this setup, a maximum of two switching events in one time-step are considered ,therefore, it is assumed that these vectors have a maximum size of two.

In Fig. 5.13, at the first time-step, there is no *State* transition and therefore, *States* are reported as [-1 -1]. If there is no transition the *Time-stamps* output is [Ts Ts], where Ts is the value of time-step. The status is [0] which indicates no error took place during the time-step. In the next time-step, the input signal became high producing *Sate* as [1 -1], *Time-stamps* as [T1 Ts] and the *Status* as [0]. This can be interpreted as one transition happened and the signal became high. Again no error is indicated by the *Status* signal as it displays [0]. In the third time-step, the *State* is [0 1], the *Time-stamps* is [T1 T2] and the *Status* is [0]. In this time-step, two transitions took place. In fourth and fifth time-step, no transition and one transition took place respectively, and as usual no error signal is shown. However, in the sixth time-step three transitions took place and the *Status* displayed an *error* message. The *State* and the *Time-stamps* output reported only the first two transitions as that is the maximum allocated size of the vectors. If any error message appeared, the simulation would continue but with errors in the results.

Using the *State* and the *Time-stamps* information from the event detector block, the s-function performs the necessary computations of system equations to produce the voltages, currents, powers and other signals. Only eight signals/channels are sent to the controller for the generation of the gate pulses for the next time-step and the other four signals are used for monitoring purposes. The number of monitoring channels can be increased or decreased as required, however, it will impact the minimum achievable time-step for the HIL simulation. The developed SIMULINK model implementing the s-function is given in Appendix C.

5.6.5 Building of the S-function

The general structure of the s-function written in C can be depicted through the flowchart shown in Fig. 5.12(b). The s-function is invoked at each time-step and any changes in the s-function inputs can be passed to it even while the simulation is running. For the HIL simulation of the VSC-HVDC, the system receives six gating signals from the controller which are fed to the RTLAB *Event Detector* block.

The s-function shown in Fig. 5.12 receives twelve signals from the event detector block as input and it produces another twelve signals as output. Out of twelve output signals, eight are fed-back to the controller and the rest are used for monitoring purposes. The first block in the s-function flowchart is implemented through the function *mdlInitializeSizes* which is used to specify the basic characteristics of the s-function such as number of inputs and outputs, port width of each input and output, and the number of parameters used. It is worthwhile to mention that while the simulation is running, the s-function parameters cannot be changed but its inputs can be changed.

The second block is the assignment of sample time or time-step of the simulation which is done through retrieving a function *mdlInitializeSampleTimes.* In the third step, through a function *mdlStart*, all variables are initialized, data files are read, and pre-computed variables and matrices are read. *mdlOutputs* is the next function which is called at each time-step. All the outputs requested are listed here and produced at each time-step. The next block is *mdlUpdate*, also implemented at each time-step. The main program is placed under *mdlOutputs* functions which performs all the necessary computations. It also updates any changes that takes place to the inputs while the simulation runs. For the VSC-HVDC, all the voltages and currents are calculated and some of them are monitored on the oscilloscope. In addition to producing the real-time outputs through I/Os, data can also be stored in a data file for off-line use. Finally, when the simulation is completed, *mdlTerminate* function is invoked to indicate the end of the simulation. At this stage, all the memory blocks are freed to ensure no memory leakage.

5.7 Experimental Setup

In order to validate the simulation results a 4kW VSC-HVDC system is built in the laboratory. Fig. 5.14 shows the schematic of hardware experimental setup. The device data for the experimental setup is the same of HIL simulation data. The ac-supply is the three-phase $117V_{l-n}$ supply available in the lab. The same supply is used on both sides of the VSC-HVDC system as it is also the same in the HIL simulation.

Three-Phase Variac

A three-phase variac was added in the setup so that the ac voltage can be applied gradually at the start of the experiment when gating pulses are not yet applied to the IGBTs. This protects the capacitor from blowing up due to the inrush current if full voltage is applied suddenly. The variac is rated at $208V, 20A$, and it can produce an output voltage in the range of $0 - 208V$ line-to-line.

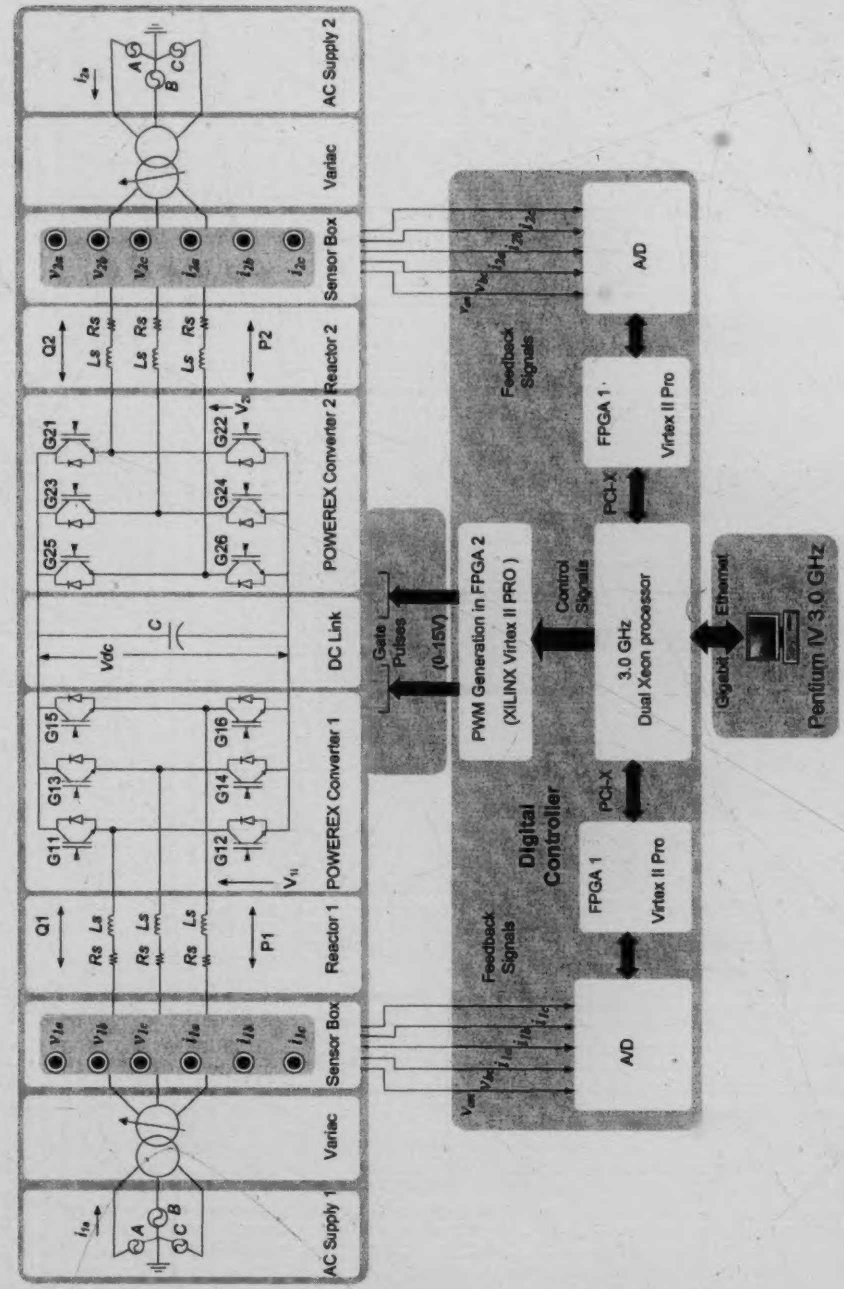


Figure 5.14: Experimental setup for the VSC-HVDC system.

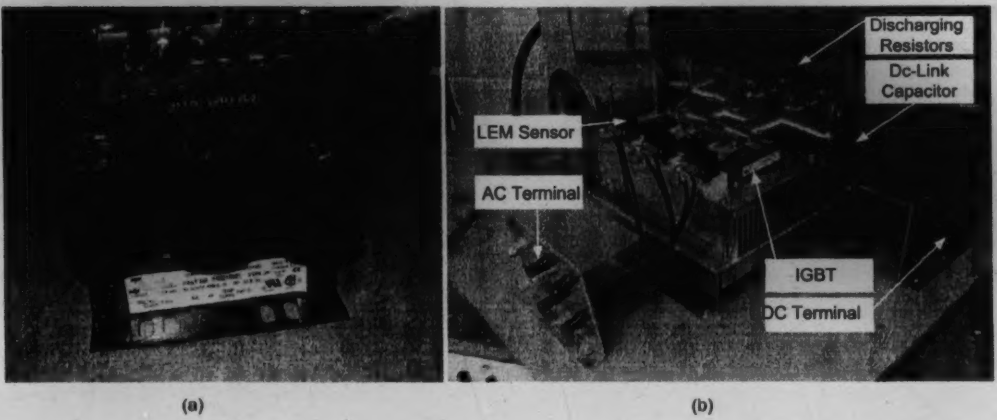


Figure 5.15: Experimental Hardware: (a) reactor (b) converter used in the experiment.

Series Reactor

A $2.5mH$, 600V, $18A$ ac inductor as shown in Fig. 5.15 was used. Due to the low rated inductor, the *rms* of ac currents must not exceed the $18A$.

Pow-r-Pak Converters

The Pow-r-Pak is an IGBT based three-phase converter power assembly that can be used for various purposes such as motor control, power supply, UPS or other power conversion applications. Similar type of converters but with higher ratings are used in the industry for STATCOM, UPFC and VSC-HVDC systems. The Pow-r-Pak converter model used for the experimental setup is suitable for operation with DC bus voltages up to 800VDC and switching frequencies up to 20kHz. Detailed information about the Pow-r-Pak module and its components are available in the manufacturer's web-site.

Two Pow-r-Pak converters are connected in parallel to construct the VSC-HVDC circuit as shown in Fig. 5.14. Each converter consists of six IGBTs, a gate drive board with in-built protections, LEM sensors to measure currents, and a large heat-sink with a blower for cooling the IGBTs. The gate drive board is mounted on top of the module and it implements a default $2\mu s$ dead-time so that DC-link never becomes short-circuited. Among the many protection features of the converter, over current protection, over-voltage and under-voltage protection, excessive temperature protection are noteworthy. Fig. 5.15(b) shows the snapshot of the converter and Fig. 5.16 shows the entire setup of the experimental hardware for the VSC-HVDC system.

Sensor Boxes and Voltage Probes

Two sensor boxes and a few differential voltage probes are used for measuring various voltages and currents in the experimental setup. The sensor box are built in the lab and they use LEM voltage and current sensors to produce the instantaneous voltages proportional to the currents. They are used for measuring the supply voltages and currents in the line which are fed back to the controller. Sensor boxes showed high fidelity over an acceptable range of frequencies, however, they produce some dc offset which needed to be compensated in the control algorithm. Each sensor box is capable of measuring three voltages and four currents. In addition, a few more differential voltage probes are used for measurement and monitoring purposes.

Digital Controller

Similar to the HIL setup the *Target* 2 node in the PC-cluster is used as the digital controller where the same control scheme is implemented. Analog feedback signals ($\pm 15V$) measured through sensor boxes and probes are sent to the controller and digital gating signals (0-15V) are sent to the IGBT gate driver module. The entire experiment was monitored and controlled from the *Host* 2 computer.

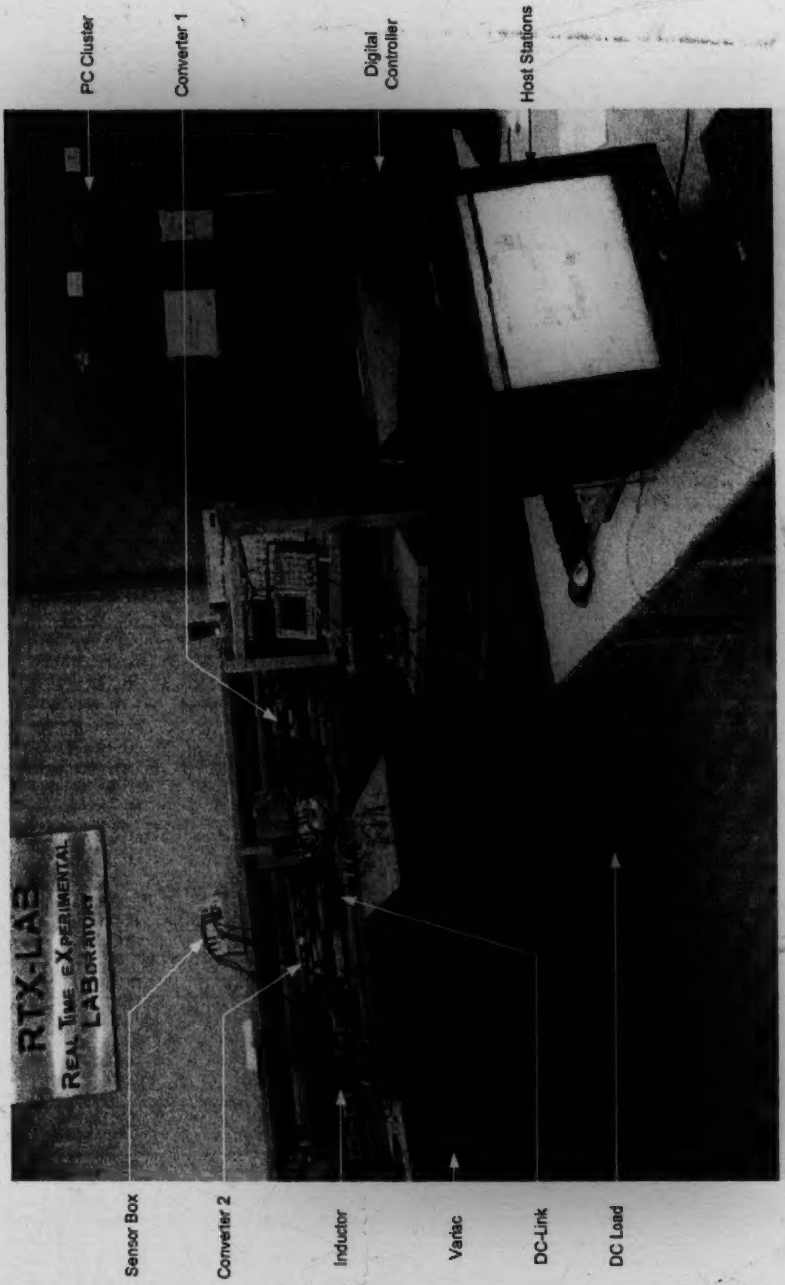


Figure 5.16: Snapshot of the hardware setup for the VSC-HVDC experiment.

5.8 Results and Discussion

Three sets of results from: (1) off-line C program (2) HIL simulation (3) experimental validation, are collected and compared for each case study. Studies are conducted for both steady-state and transient situations. For the off-line simulation using the C program, a $20\mu s$ time-step is used and the data collected at every time-step. Therefore, to avoid a large volume of data, only one second data was recorded. However, for experimental results and HIL simulation results, data was collected at every $250\mu s$. Due to the involvement of hardware which takes approximately $10s$ to initialize the system, the experimental system needed to run for a much longer time than off-line simulation. Moreover circuit breaker switching operations are done manually which needed extra time. This is the reason for data being collected at a $250\mu s$ interval.

5.8.1 Steady-State

For steady-state results, the initial start-up transients are excluded. Two cases have been considered for all three categories and closed-loop results are given.

Case 1 Power references are kept positive

In this case, all power reference quantities are set to positive values as given below:

- $V_{dc} = 500V$ and $P_2 = 3000W$
- $Q_1 = 2000\,VAr$ and $Q_2 = 2000\,VAr$

Fig. 5.17 shows the steady-state results obtained through the off-line simulation, and Fig. 5.18 and Fig. 5.19 show the real-time oscilloscope traces obtained from HIL simulation and the experiment, respectively. In most cases, the results are in agreement. A mismatch between DC voltage is observed which is mainly due to the voltage probe. In the experimental case, the dc-voltage has been measured using a DC voltmeter and have been found in the range of $497V$ to $502V$, even though the oscilloscope trace shows it slightly low ($482V$). This mismatch is due to the

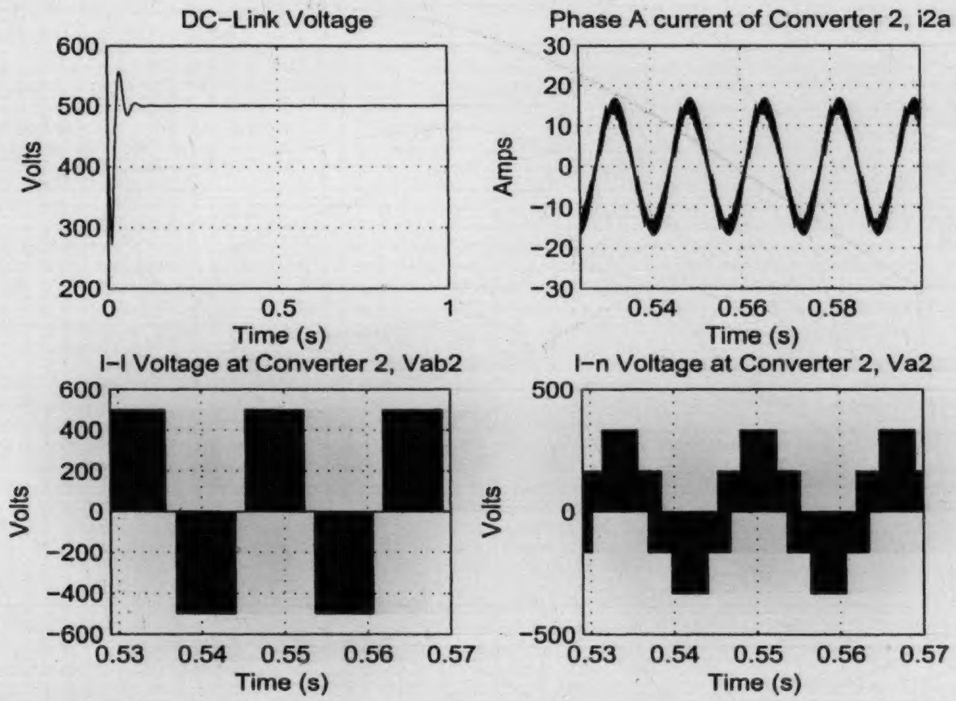


Figure 5.17: Off-line simulation results of steady-state DC-link voltage, V_{dc}, phase-a current, i_{a2}, line-to-line voltage, v_{ab2}, and phase voltage, v_{a2}.

differential probe used for oscilloscope measurement. The phase voltage and line-to-line voltage for all three cases are similar and as expected. The peak line-to-line voltage and line-to-neutral voltage are found slightly higher in the HIL simulation and the experimental results due to the presence of spikes in the voltages when the transition of voltages from one level to other takes place. The current waveform looks similar in all cases except in the experimental result, it is a little distorted sinusoidal signal. This is due to the distorted supply voltage in the laboratory. The frequency of the current is found to be $59.95Hz$ for off-line simulation whereas, for both HIL simulation and experimental results, they are found to be $60Hz$. The fundamental component and rms values are $11.92A$ and $13.74A$ for off-line simulation, $12.6A$ and $13.8A$ for HIL simulation and $11.9A$ and $13.5A$ for the experimental results, respectively.

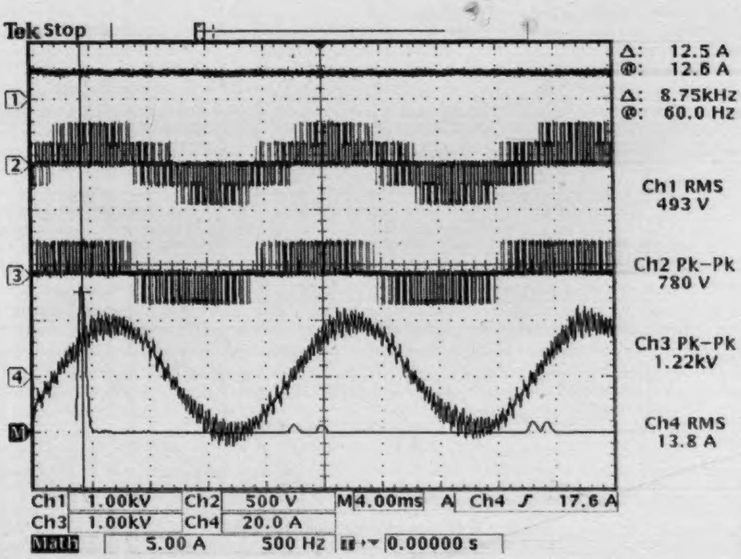


Figure 5.18: Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, V_{a2}, line-to-line voltage, V_{ab2}, and phase-a current, I_{a2} and FFT of the current I_{a2} for HIL simulation.

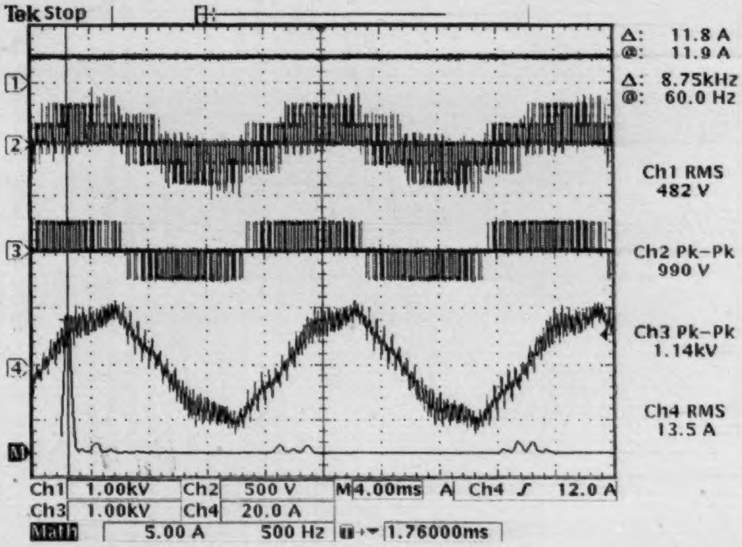


Figure 5.19: Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, V_{a2}, line-to-line voltage, V_{ab2}, and phase-a current, I_{a2} and FFT of the current I_{a2} obtained from the experiment.

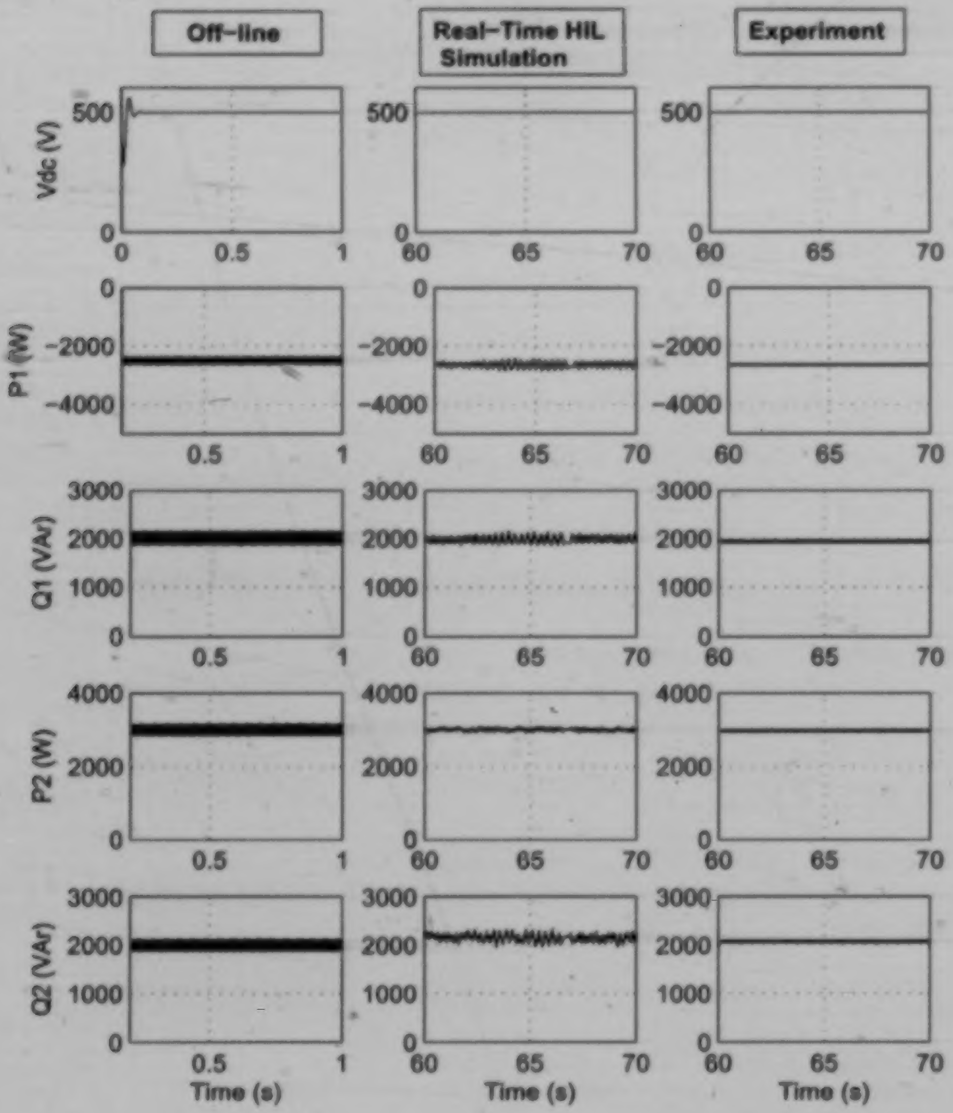
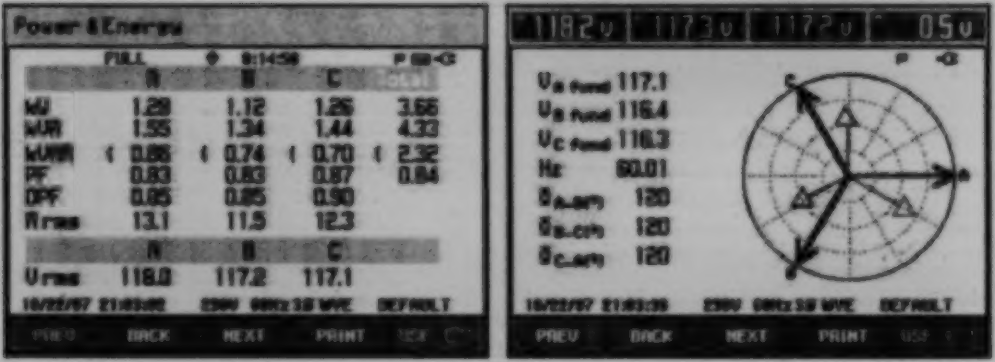


Figure 5.20: Off-line Simulation, HIL simulation and experimental results of steady-state DC-Link voltage, V_{dc}, VSC 1 power, P_1 and Q_1, and VSC 2 power, P_2 and Q_2 for the case when all reference powers are positive.

In Fig. 5.20, the DC-Link voltage, the active and reactive powers at both ends are shown. All of them have been found to follow the reference quantities. However,



Power & Energy

	A	B	C	Total
kW	1.28	1.12	1.26	3.66
kVA	1.55	1.34	1.44	4.33
kVAR	0.86	0.74	0.70	2.32
PF	0.83	0.83	0.87	0.84
DPF	0.85	0.85	0.90	
Arms	13.1	11.5	12.3	

	A	B	C
Vrms	118.0	117.2	117.1

(a)

118.2 V
117.3 V
117.2 V
0.5 V

$V_{A\,fund}$ 117.1
$V_{B\,fund}$ 116.4
$V_{C\,fund}$ 116.3
Hz 60.01
$\theta_{A\text{-}B}$ 120
$\theta_{B\text{-}C}$ 120
$\theta_{C\text{-}A}$ 120

(b)

Figure 5.21: Fluke measurements from the experiment (a) Power, voltage and current (b) Phasor diagram.

the VSC 1 power, P_1 which is not a reference quantity but flows due to the active power demand, P_2 from VSC 2 shows slight mismatch. In fact, VSC 1 power is not controlled and should be determined by the reference power, P_2 plus losses. The negative sign of P_1 indicates that power is flowing against the assumed direction. In this case, for a 3000W power at converter 2, the average power measured at the converter 1 AC side is found to be $-2595.62W$ in off-line simulation, $-2652.35W$ in HIL simulation and $-2696.5W$ in experimental result. This indicates that the total loss in both ac and dc side is $404.38W$ (off-line simulation), $347.65W$ (HIL simulation) and $303.5W$ (experiment). Both off-line and HIL simulation results are found to be close to experimental results other than minor differences which are mainly due to the differences in the currents computed or measured.

In Fig. 5.21(a), the measurements at the terminal of the ac supply 2 using the Fluke three-phase power quality analyzer are shown. It has been found that both real and reactive power are slightly higher than that are found through the calculation in the controller. This is due to the initial difference of power levels when no power flow is taking place. It was found that even for zero reference power situation, the power meter shows some positive offset values (0.3 kW and 0.67kVAr) which is always there. The reason could be due to the phase-lag introduced in the signal when they are taken back to the controller. Another minor discrepancy is observed in case of currents. The currents are slightly unbalance which is due to the unbalance in supply voltage and unbalance in the value of impedances offered by each line.

Case 2 Power references are negative

In this case, all power references are set to negative value as given below:

- $V_{dc} = 500V$ and $P_2 = -4000W$
- $Q_1 = -2000\,VAr$ and $Q_2 = -2000\,VAr$

A negative active power means the direction of the power flow is in the opposite direction. Reactive power could be inductive or capacitive: it depends on whether the converter is absorbing or producing the reactive power. Fig. 5.22 shows the steady-state results obtained through the off-line simulation (C-program), and Fig. 5.23 and Fig. 5.24 show the oscilloscope trace for the HIL simulation and the experiment. The results are in agreement with the fundamental components of current being 14.06A (off-line simulation), 12.8A (HIL simulation) and 12.5A (experiment). The DC voltage in the oscilloscope results are slightly less as observed in the previous case which is due to the differential probe offset. The line-to-line voltages and the phase voltages are in close agreement neglecting the minor discrepancy in their values.

The DC voltages and calculated powers at both converters are shown in Fig. 5.25 for all three cases. DC voltage is found to be smoothly controlled and maintained at 500V. All the powers do match except slight mismatch at VSC 1 active power, P_1. The average P_1 is found to be 4912.3W (off-line), 4345.6W (HIL simulation) and 4432.6W (experiment). This indicates that 4000W power is supplied to the ac side 2 of the grid through VSC 2 but VSC 1 drew slight higher power from the other side of grid. The difference between P_1 and P_2 is the power lost in the process. The off-line simulation shows the maximum power loss in the system.

Fig. 5.26 shows the measured power through the power quality analyzer. It is found that the measured power at VSC 2 terminal is less than the power calculated in the controller. This is again due to the power offset which shows some positive power even the the controller shows zero power. The power calculation at the

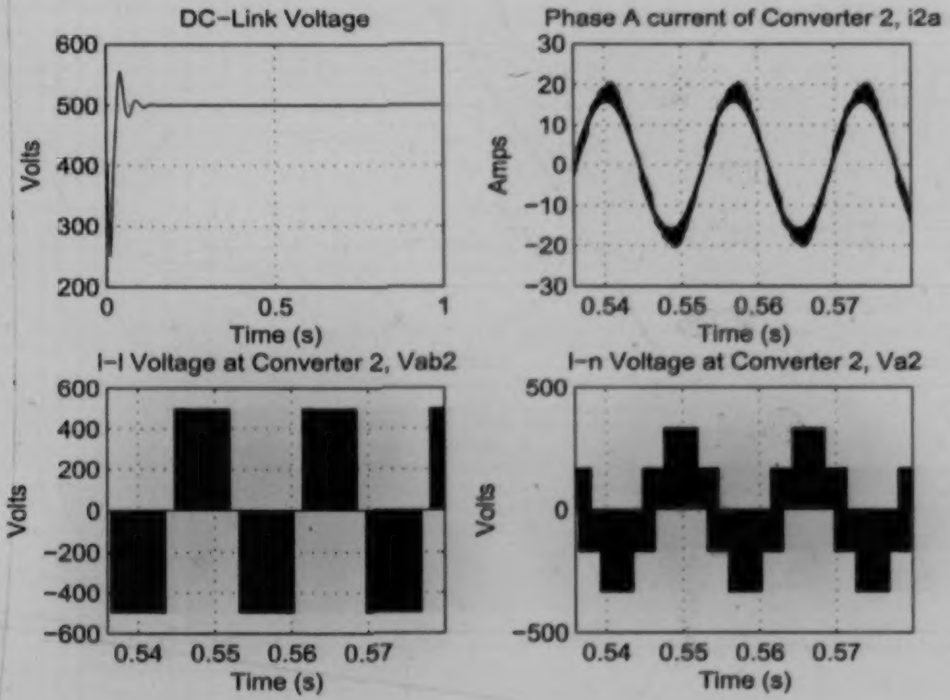


Figure 5.22: Off-line simulation results of steady-state DC-Link voltage, V_{dc}, phase-a current, i_{a2}, line-to-line voltage, v_{ab2}, and phase voltage, v_{a2}.

controller is performed from the voltage and currents received from the sensor boxes. These voltages and currents are subject to some delays during the feedback process. Moreover, to make the control effective, currents are needed to be filtered for a running average current instead of the actual current with noises and oscillations. These two are the main reasons behind the mismatch between the calculated power from the controller and the measured power from the Fluke power quality analyzer.

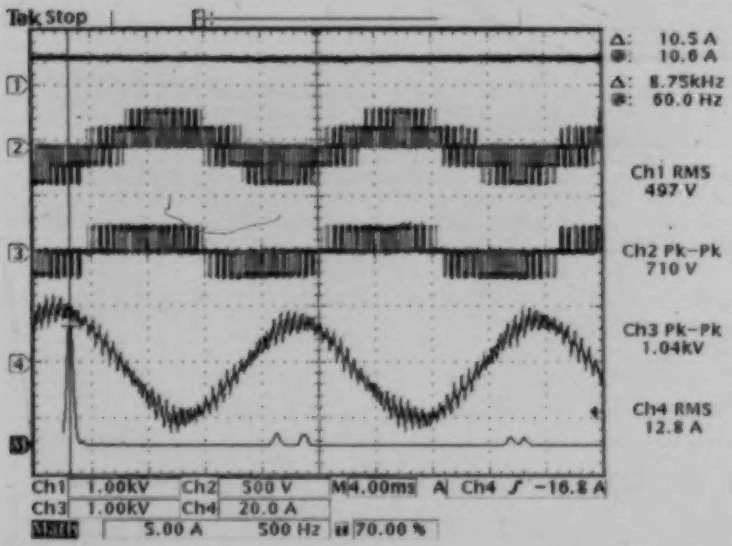


Figure 5.23: Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, v_{a2}, line-to-line voltage, v_{ab2}, and phase-a current, i_{a2} and FFT of i_{a2} for HIL simulation.

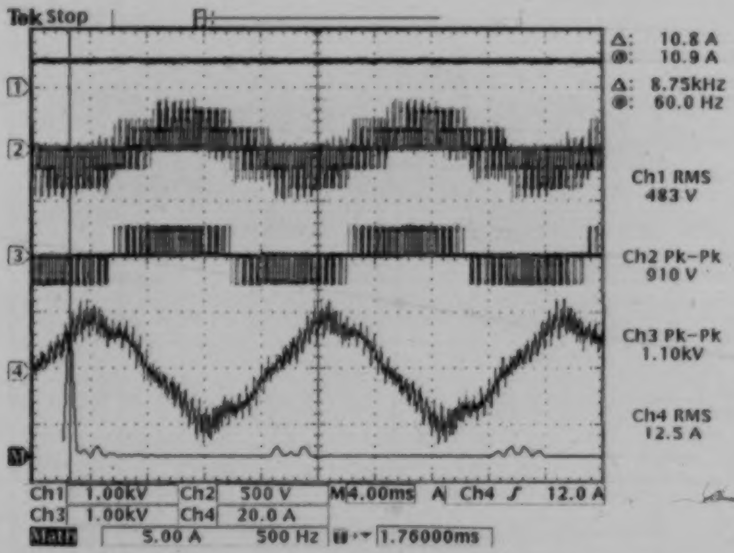


Figure 5.24: Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, v_{a2}, line-to-line voltage, v_{ab2}, and phase-a current, i_{a2} and FFT of i_{a2} obtained from the experiment.

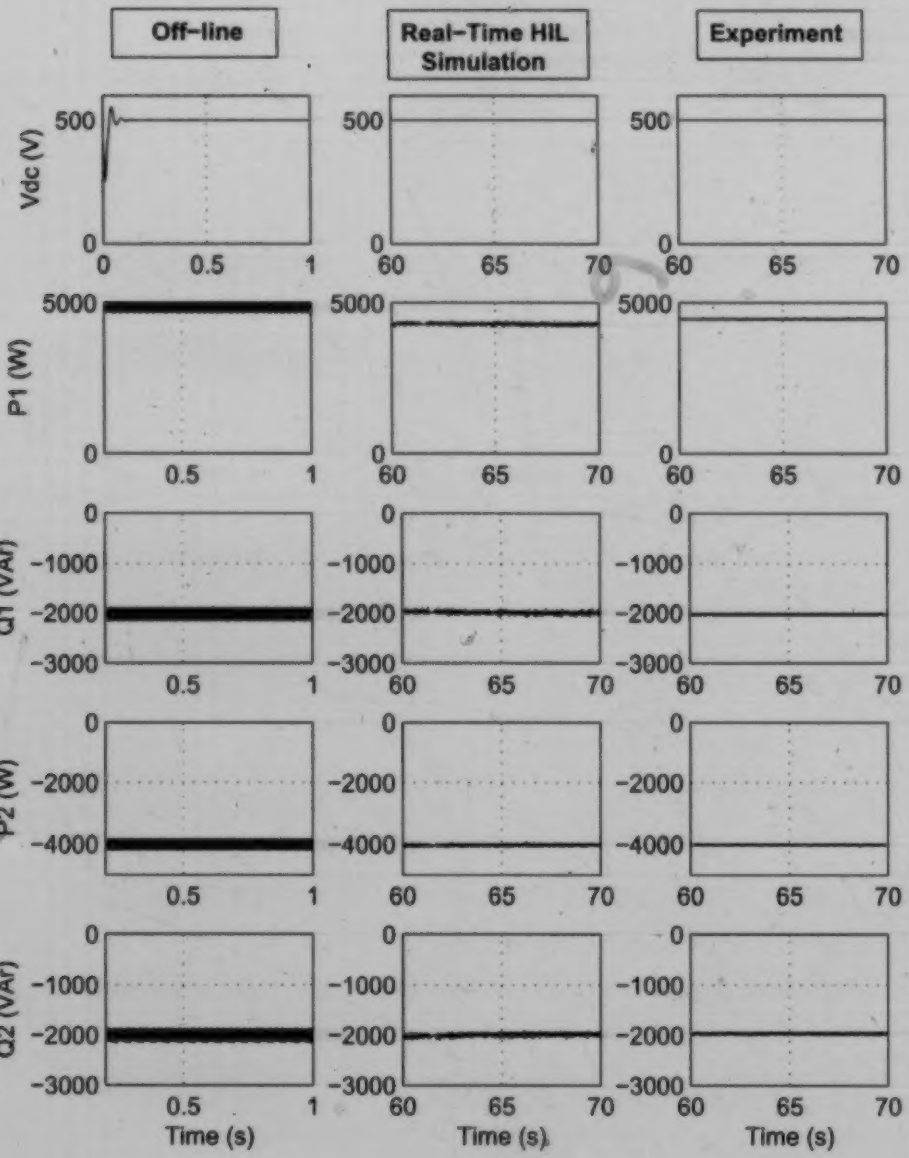
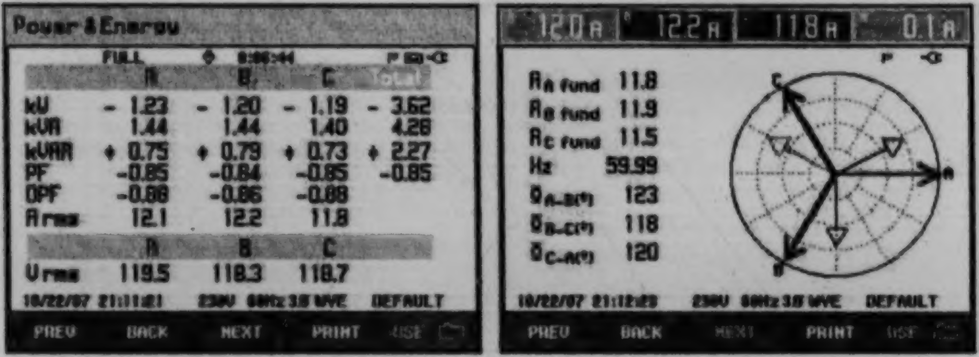


Figure 5.25: Off-line simulation, HIL simulation and Experimental result of steady-state DC-Link voltage, V_{dc}, VSC 1 power, P_1 and Q_1, and VSC 2 power, P_2 and Q_2.



Power & Energy

FULL 0:06:44

	A	B	C	Total
kW	- 1.23	- 1.20	- 1.19	- 3.62
kVA	1.44	1.44	1.40	4.26
kVAR	♦ 0.75	♦ 0.79	♦ 0.73	♦ 2.27
PF	-0.85	-0.84	-0.85	-0.85
DPF	-0.88	-0.86	-0.88	
Arms	12.1	12.2	11.8	

	A	B	C
Vrms	119.5	118.3	118.7

10/22/07 21:11:21 230V 60Hz 3Ø WYE DEFAULT

12.0 A 12.2 A 11.8 A 0.1 A

$A_{A\,fund}$	11.8
$A_{B\,fund}$	11.9
$A_{C\,fund}$	11.5
Hz	59.99
$\phi_{A-B}(°)$	123
$\phi_{B-C}(°)$	118
$\phi_{C-A}(°)$	120

10/22/07 21:12:23 230V 60Hz 3Ø WYE DEFAULT

Figure 5.26: Fluke measurements from the experiment for negative reference power (a) Power, voltage and current (b) Phasor diagram.

5.8.2 Effect of Distorted Supply Voltage on Steady-State Results

From Fig. 5.19 and Fig. 5.24, it is evident that the current produced by the experimental results is quite distorted, whereas the current obtained through HIL simulation is almost sinusoidal with high frequency components in it. The main reason for the distorted experimental results is the supply voltage distortion in the lab. HIL simulation considers perfect sinusoidal waves as the source voltage whereas the supply voltage used for the experiment is quite distorted. The snapshot of the supply voltage is given in Fig. 5.27. The %THD of the supply voltage used for the experiment was found to be 5.76% and the harmonic components of the supply voltage are listed in Table 5.1.

To compare the HIL simulation results with the experimental results it is necessary to keep system parameters and system inputs same for both cases. Efforts have been maximized to maintain system parameters as close as possible even though stray capacitances and inductances, system nonlinearity, drifting of parameters due to temperature etc. are hard to model in the simulation. However, as it is difficult to get purely sinusoidal supply voltage as an input for the experiment (due to distorted supply voltage in the laboratory), it is easier to record the distorted supply voltage and apply it in the HIL simulation as an input voltage. In order to do that, the

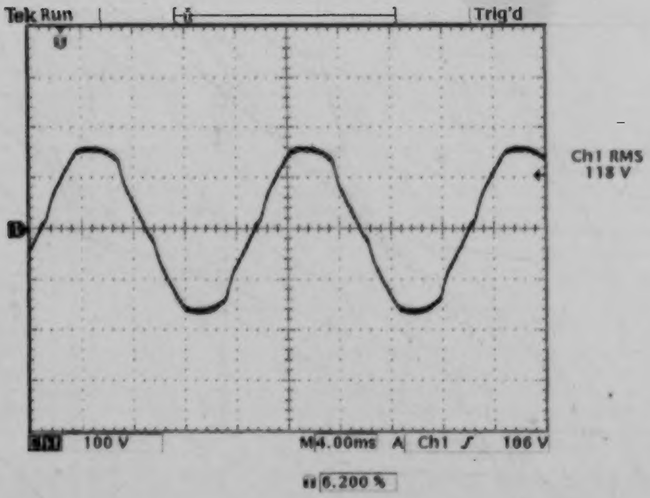


Figure 5.27: Oscilloscope trace of supply voltage in the laboratory.

supply voltage data of the lab was recorded and a Fast Fourier Transform (FFT) was performed on it to get harmonic components. Table 5.1 lists the dominant components of harmonics found in the supply voltage. The supply voltage is then reconstructed from its harmonic components and applied in the HIL simulation. Only significant harmonics are considered to reconstruct the supply voltage. The HIL simulation may still not be the same as it is in the experimental case due to the reasons mentioned above along with the exclusion of harmonic components with small value, however, the objective here is to show that distorted supply voltage plays a dominant role in producing distorted currents.

The oscilloscope trace of the steady-state results for HIL simulation using reconstructed supply voltage is given in Fig. 5.28 and Fig. 5.29. It is clear that in both cases of positive and negative power flow, the currents are distorted as in Fig. 5.19 and Fig. 5.24. Despite the fact that there is still minor differences left between the HIL simulation results and the experimental results, using the reconstructed similar supply voltage brings the two results much closer. This also confirms that distorted supply voltage is a dominant source of distorted current waveforms in the experimental results and it also validates the faithfulness of the HIL simulation.

Table 5.1: Harmonic components of laboratory supply voltage.

Harmonic Order	Amplitude
V_1	175.76
V_2	0.29
V_3	1.06
V_4	0.28
V_5	6.92
V_6	0.24
V_7	3.65
V_8	0.22
V_9	0.48
V_{11}	2.74
V_{13}	1.62
V_{17}	0.95
V_{19}	1.25

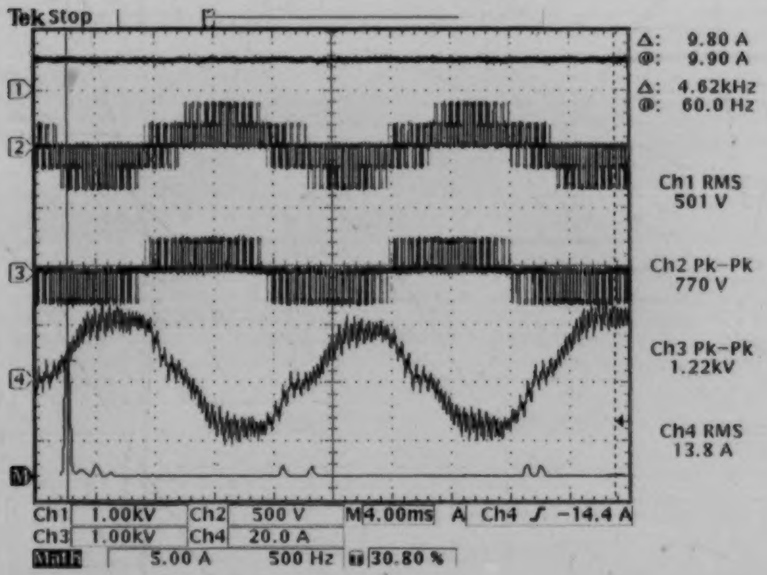


Figure 5.28: Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, v_{a2}, line-to-line voltage, v_{ab2}, and phase-a current, i_{a2} and FFT of i_{a2} for HIL simulation using reconstructed distorted supply voltage (For positive power).

5.8.3 Transients

Three cases of transient study have been performed and results are compared to investigate the responses using off-line simulation, HIL simulation and the experiment.

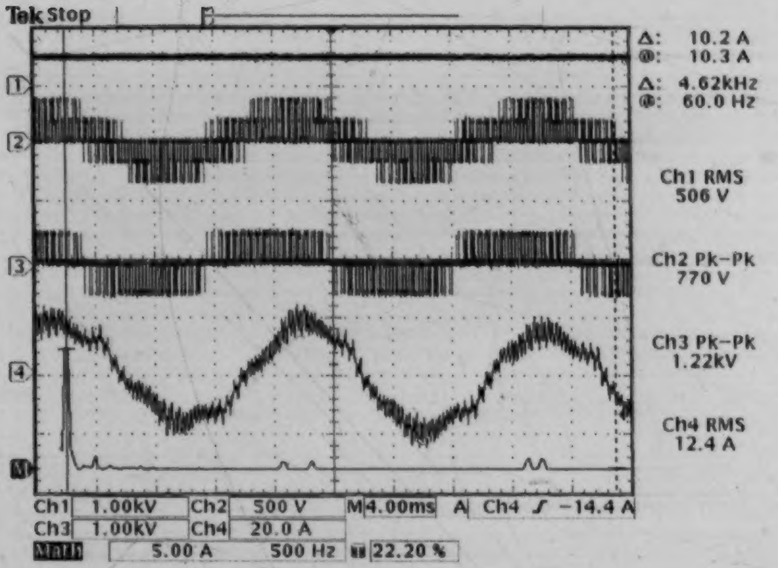


Figure 5.29: Oscilloscope trace of steady-state DC-Link voltage, V_{dc}, phase voltage, v_{a2}, line-to-line voltage, v_{ab2}, and phase-a current, i_{a2} and FFT of i_{a2} for HIL simulation using reconstructed distorted supply voltage (For negative power).

Case 1 DC voltage reference is changed while power references remained unchanged

In this case, Power references are kept at $P_2 = 3000W$, $Q_1 = 2000\,VAr$ and $Q_2 = 2000\,VAr$ while DC-link voltage reference is changed. When the system starts to run under closed-loop, the initial DC voltage reference is kept at 420V. After a while the reference voltage is changed to 560V in the first step and then to 440V in the second step.

From Fig. 5.30, it can be seen that the DC-link voltage has settled to its new values after the initial transients. There might be slight difference between the transient peaks and their settling time, however, the differences are not much and the results are acceptable. The reasons for these differences are as follows:

- The changes in references were performed at different times. Because of the huge data generated, the off-line simulation is performed for only 1s. However, for

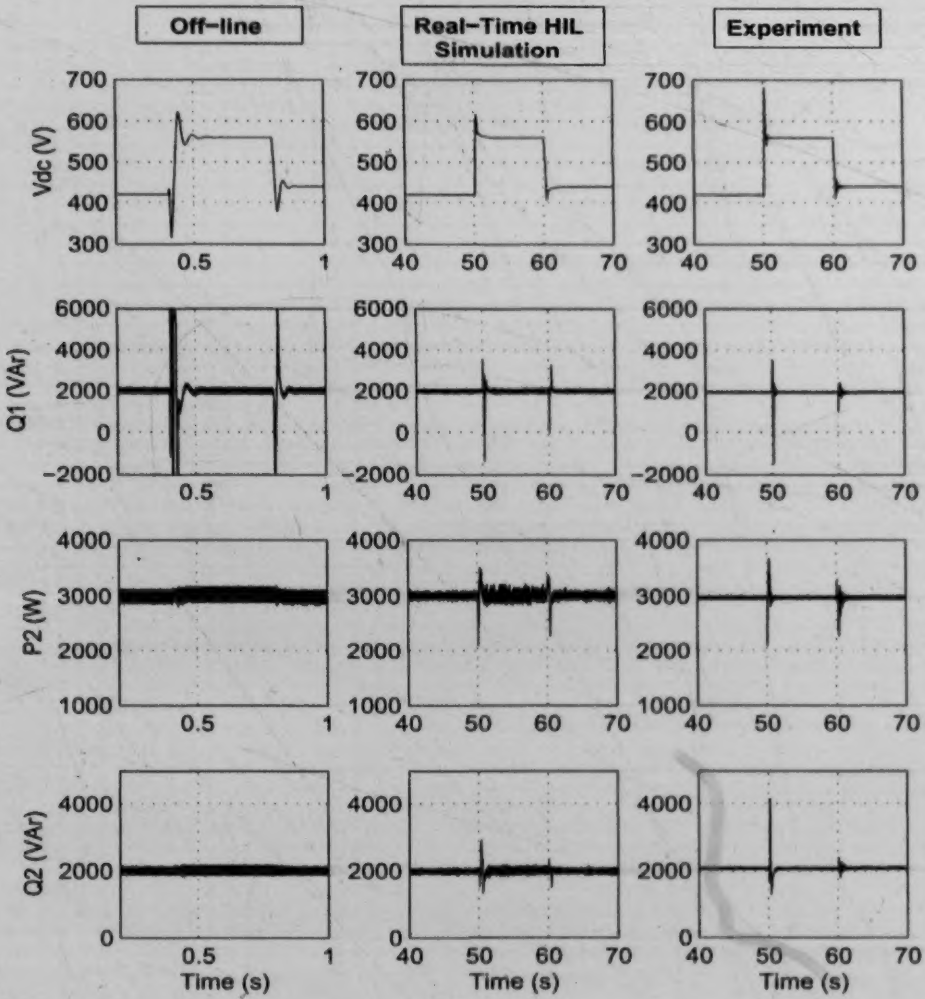


Figure 5.30: Off-line simulation, HIL simulation and experimental results of DC-Link voltage, V_{dc}, VSC 1 reactive power, Q_1, and VSC 2 power, P_2 and Q_2 for changes in the DC-link voltage reference.

HIL simulation and experiment, it takes a while (approximately 10s) to bring the system into operation because of the involvement of manual operations. Therefore, the time of changing the step in DC-voltage for all three cases could not be maintained at the same instant.

- The PI parameters found from control design was used in off-line simulation. These gains needed re-tuning when used in HIL simulation and experiment. This is due to the differences in the system parameters between the actual values and the assumed values. As a result the PI gains are slightly different for all three cases.

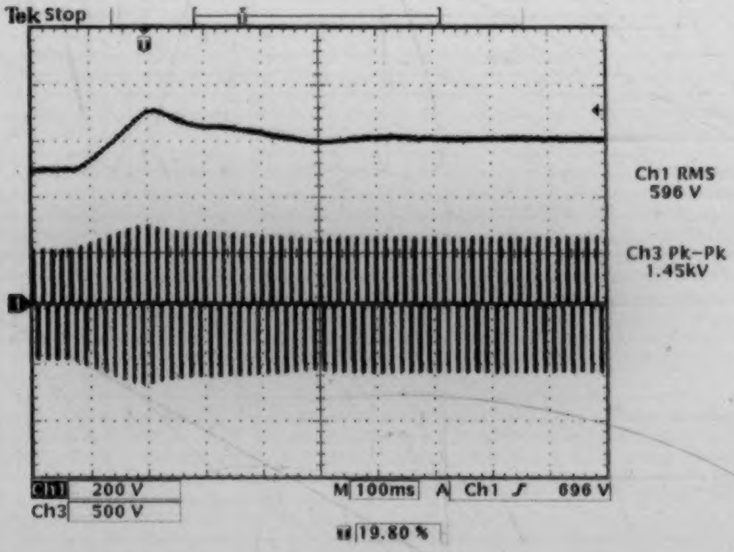


Figure 5.31: Oscilloscope trace of transient responses in DC-link voltage, V_{dc} and line-to-line voltage, v_{ab2} for HIL-simulation.

The change in DC-link voltage more or less affects all the power levels. Momentary disturbances are created in all power levels which die out very fast. Other than some minor discrepancies in the transient magnitude and response, the three cases produces stable results which are in close agreement. To be more precise, HIL-simulation result shows close resemblance with experimental results than the result obtained through off-line simulation. The transition in DC-link voltage and line-to-line voltage at the terminal of VSC2 for both HIL-simulation and the experiment are captured on the oscilloscope and shown in Fig. 5.31 and Fig. 5.32. From the responses it can be observed that for both HIL simulation and the experiment, the responses are very similar.

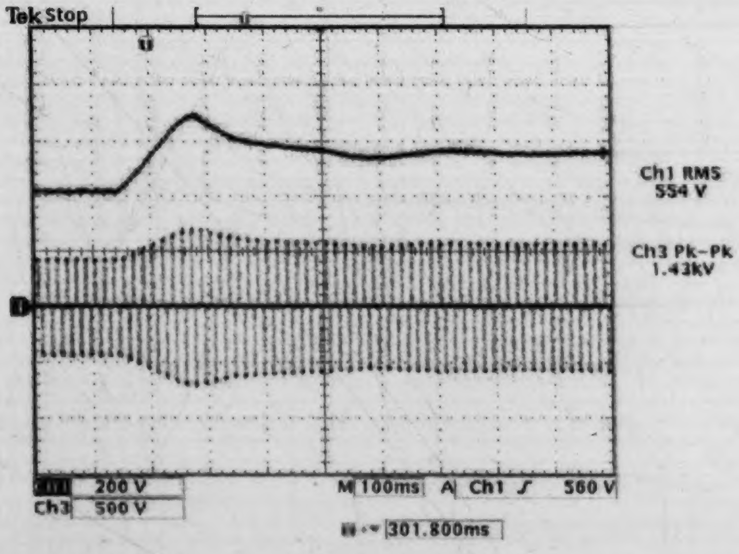


Figure 5.32: Oscilloscope trace of transient responses in DC-link voltage, V_{dc} and line-to-line voltage, v_{ab2} for HIL-simulation.

Case 2 External load is added to the DC-link

In this case, a disturbance is created in the DC-link by adding a resistive load and then disconnecting it after a while. Before the system was subject to the disturbance a steady power flow situation is reached with $V_{dc} = 500V$, $P_2 = -3000W$, $Q_1 = 2000\,VAr$ and $Q_1 = 2000\,VAr$. The responses are shown in Fig. 5.33.

A DC load of 150Ω is suddenly connected across the DC link. This caused a current flow of 3.33A through the DC load. Sudden addition of this load, however, did not have significant impact on the system. As it is evident from Fig. 5.33 that DC-link voltage remains constant all over. This concludes that as long as the DC-link voltage is controllable, any disturbance in the DC-link has minimal impact on the system operation.

Even though 150Ω load bank did not have any significant impact on the system operation, sudden impact due to a higher load creates instability in the system and the control algorithm failed to maintain the DC-link voltage. The system then collapses

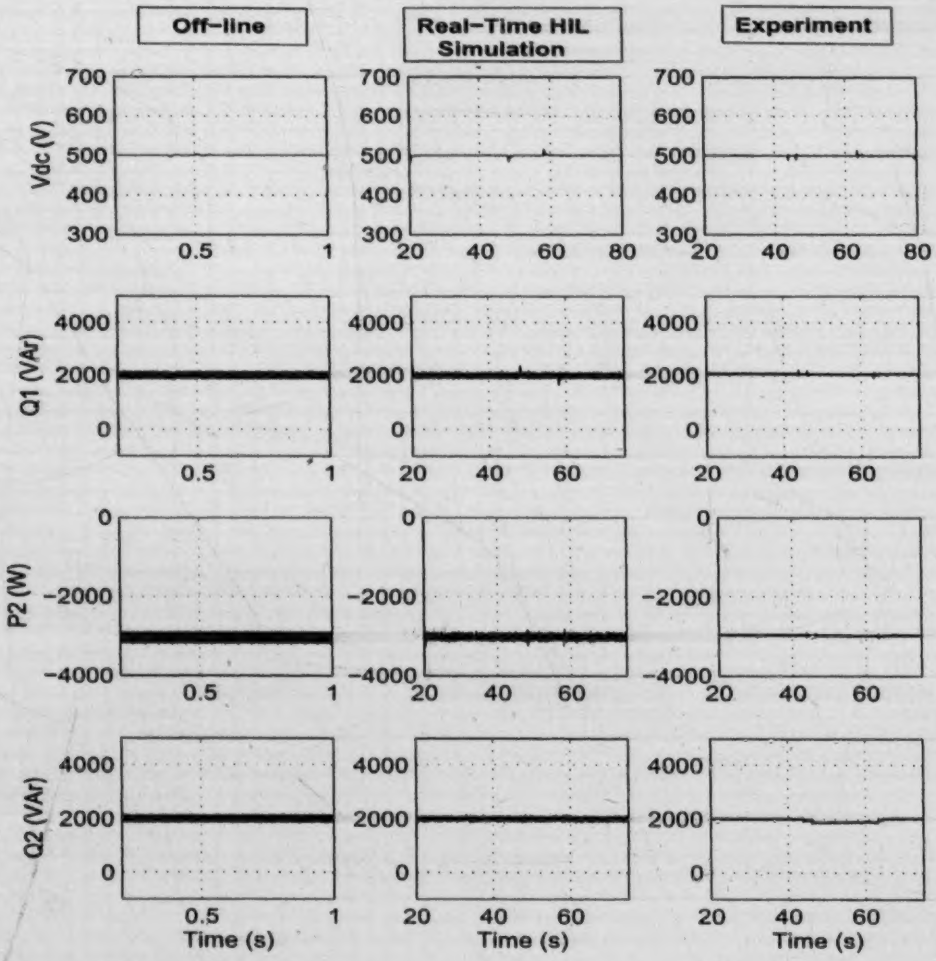


Figure 5.38: Off-line simulation, HIL simulation and experimental results of DC-Link voltage, V_{dc}, VSC 1 reactive power, Q_1, and VSC 2 power, P_2 and Q_2 when a load is added in the DC-link.

and runs in the open-loop which resembles the scenario of two floating STATCOMs sharing the same DC-link. This is due to the changes in the system parameters caused by the addition of DC-load bank for which controller gains needs to be retuned. Depending on the robustness of the control, the system is able to maintain its operation for an addition of small percentage of load but will certainly loose control if the load is large enough to change the system's characteristic by a big margin.

Case 3 Power references are changed while DC-Link voltage remains unchanged

Here all the three power references went through multiple changes including power reversal while the DC-link voltage is kept unchanged. The responses are shown in Fig. 5.34 where it has been found that DC-link voltage remains almost unaffected even though power demands went through multiple changes. Small spikes are observed in the experimental results and HIL simulation which are due to the changes of various power references. However, off-line simulation does not exhibit any spike or disturbances in the DC-link voltage. For changes in power levels, transients are observed in experimental results and then HIL-simulation also shows some transient spikes when power levels are changed. It can be observed that in the case of HIL-simulation and experiment, while a change takes place in any of the power references, slight disturbances are observed in the rest of the power levels, however, off-line simulation seems to show immunity in those circumstances.

From all these case studies, it can be concluded that HIL-simulation is more realistic and in close agrement to the experimental results than the off-line simulation. In most cases, they showed almost the same results which validate the accuracy of the HIL simulation. However, minor discrepancies are observed due to the following differences or limitations:

- For both off-line simulation and HIL simulation, the source voltage is considered to be perfectly sinusoidal and balanced whereas in the experiment, the supply voltage was slightly unbalanced and distorted which is further worsened when it was taken through the variac.
- The amplitude of the supply voltage in off-line simulation and HIL-simulation is assumed constant. It does not change its amplitude based on the mode of operation of the VSC converters. However, in the case of experiment, slight

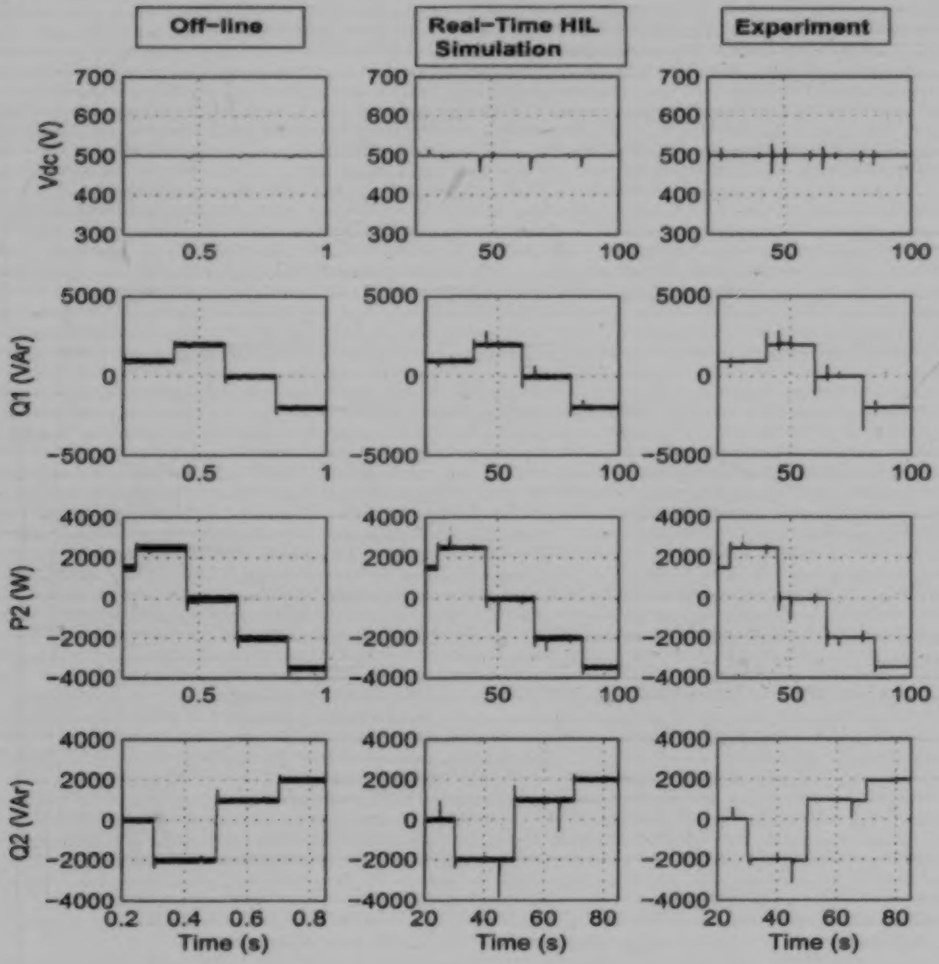


Figure 5.34: Off-line simulation, HIL simulation and experimental result of DC-Link voltage, V_{dc}, VSC 1 reactive power, Q_1, and VSC 2 power, P_2 and Q_2 when power references were changed.

variation is observed between the inductive mode of operation and the capacitive mode of operation of the VSC. It has been observed that the voltage is increased slightly when the VSC supplies reactive power to the grid and vice versa.

- Power measurements are performed in the *Target* that runs control algorithm. Target receives voltage and current signals through differential voltage probes and sensor boxes. These probes and sensor boxes contribute errors in the measurements due to the delay in the response time. In addition to that, all the channels in the sensor boxes and the differential voltage probes together with I/Os of the computers produce dc-offsets of various degrees. Attempts had been taken to minimize the dc-offsets by adding or subtracting a small dc value in the received signal.

- There might have been a difference between the values of system parameters used for the simulation and the values of actual parameters in the system. The actual parameters may even vary with the temperature which was not taken into account in the simulation. Moreover, non-linearities and some protection circuitry such as di/dt and dv/dt protection present in the real hardware was not considered in the simulation.
- There were unbalance in the systems parameters. Measurement showed that the inductances in three phases varied by a margin of $2-3\%$ among themselves, whereas for simulation they were considered same for all three phases.
- For implementation of the control, in the HIL-simulation and the experiment, voltages and currents are passed through low pass filters to get a running average of their values, however, for off-line simulation no filters were used.

5.9 Execution Time for HIL Simulation

Efforts have been made to optimize the model so that time-step can be brought down to as small as possible without causing overruns. The number of inputs and outputs are kept to minimum so that CPU interruption is also minimum. No monitoring output is taken from the simulator *Target*, instead signals are monitored in the controller which uses a large time-step ($250\mu s$). Applying Algorithm 2 (Chapter 2) for switching correction and implementing case specific TSSIT algorithms for transients due to sudden input changes, the HIL-real-time simulation was achieved with as low as $10\mu s$ time-step. However, if no correction algorithm is used the HIL-simulation was found as low as $10\mu s$. Without applying the correction algorithms, HIL simulation fails if the assigned time-step is greater than $26\mu s$. At $26\mu s$ time-step, the simulation works but with poor results. However, applying Algorithm 2, the simulation was possible even up to a time-step of $50\mu s$. Beyond that the results are not satisfactory.

A detailed breakdown of the time-step reveals that with correction, the total computation time was $3.985\mu s$ whereas, without switching correction it was $3.027\mu s$. Apparently, an idle time of $4.36\mu s$ is found inside the $10\mu s$ time-step which might indicate the possibility of further reduction in time-step. However, in such a case, during the dynamic changes of input/output references, at some time-step, the computation time will be higher and may exceed the assigned time-step thus causing overruns. Detailed breakdown of the time-step is given in Table 5.2

Table 5.2: Execution times for individual tasks within one time-step of $10\mu s$.

Task	**Execution Time (μs)**
Computation Time	3.985333
Idle Time	4.362667
Data Acquisition	1.210000
Status Update	0.117333
Target Request Handling	0.040000
Host Request Handling	0.035667
Synchronization Handling	0.033333
Others	0.215667
Total Step-Size	10.000000

5.10 Summary

In this chapter, an off-line simulation using the C language, a real-time HIL simulation where the system was simulated through the interfacing of a digital controller with it and an experimental validation of a VSC-HVDC system have been carried out. The main contributions of this chapter are summarized as follows:

- The detailed modeling of the VSC-HVDC system is described and is used for both off-line and real-time HIL simulation.
- Discretization methods such as RIT and TSSIT along with the switching correction algorithms are used to develop the off-line and real-time simulation program.

- A brief introduction of the PC-cluster is given. The cluster was used for both the HIL simulation and the experiment.

- A digital control scheme is implemented in a Target node of the PC cluster which uses a general purpose computer. The node has capability to interface with the external hardware. A reconfigurable carrier board is used for PWM implementation inside an FPGA. The implementation is done using SIMULINK and Xilinx XSG toolboxes. The same controller is used for both the HIL simulation and the experiment. For, HIL simulation analog and digital signals are interchanged with the simulator at every time-step.

- A 4 KW VSC-HVDC experimental setup is built by using three-phase IGBT modules. The system was operated under maximum allowable limit.

- Various case studies were performed using off-line simulation, real-time HIL simulation and the experiment. Steady-state and transient situations are studied using all three methods and close agreement among their results has been observed.

Chapter 6

Conclusions

Real-Time HIL digital simulation is increasingly being used for testing new power electronic equipment applied in power systems. HIL simulation can be used to improve performance, efficiency and to reduce the cost of field installation of such equipment. The advancement of computer technology, availability of highly efficient, low cost computing engines and easily available off-the-shelf systems, expedited the viability and usability of the HIL technology in the power industry. Real-Time HIL simulation provides significant advantages with closed loop simulation and testing of completely integrated software and hardware in real-time.

A VSC-HVDC system has been chosen in this thesis to perform the HIL-simulation as this system offers a fair amount of complexity due to its amalgamation of power system, power electronics, control and digital technology. After a literature review on the existing state of the research in this arena, the thesis identified the potential obstacles in performing accurate HIL simulation of the VSC-HVDC system and proposed practical solutions to alleviate those problems.

The contributions of this thesis are summarized as follows:

1. Accurate discretization of electrical networks is one of the main requirements for performing an electromagnetic transient simulation. Among existing methods, Trapezoidal Rule is the a widely used technique, however, in some situations, Trapezoidal Rule shows spurious oscillation which is not a reflection of the true transients. To alleviate this problem, a better and accurate discretization

method is proposed in this thesis. Time-shifted step-invariant transformation and ramp-invariant transformation are the two methods used for both off-line and HIL simulation. These methods are initially tested on simple networks such as RL and RLC networks at different conditions and excellent results are observed. Both discretization techniques can be applied to the solution of networks using transfer function based approach and state-space approach.

2. Discrete event simulation has always been a challenge in real-time simulation. In the past, several techniques were proposed for off-line simulation, however, for real-time simulation there were not many available. Moreover, most of the solutions were proposed for single switching events even though multiple switchings are not uncommon in systems such as FACTS and HVDC. This thesis proposed a family of algorithms that tackles multiple switchings in one time-step for both real-time and off-line simulation. All of the algorithms are first applied in the off-line simulation of a PWM VSC and a comparative study has been conducted. The study showed clear benefits of using the algorithms and a comparison of errors, THDs and the maximum simulation time taken by each algorithm to compute one time-step are also given.

3. As the VSC-HVDC needs a controller that produces the necessary gate pulses to generate PWM for the operation of VSC-HVDC, design and development of a fast, accurate and efficient control scheme is therefore essential. In this thesis, a control scheme is developed which is capable of facilitating the transfer of bidirectional power through the VSC-HVDC. The design of the controller is done in the discrete-domain. To verify the effectiveness of the controller, the two well-known simulation packages PSCAD/EMTDC and MATLAB/SIMULINK were used. The complete VSC-HVDC system and its controller were modeled in both software environments and off-line simulation was performed for both steady-state and transient situations. close agreement between their results validated the effectiveness of the controller. While doing this comparative study, a clear picture about the pros and cons of the two simulation tools was found.

4. The complete VSC-HVDC system was modeled using discretization techniques

described earlier. Switching correction algorithms for multiple switchings were applied, and the designed control scheme was used to develop an off-line simulation program using C language. The objective of performing another off-line simulation was to make sure that the developed model and the applied switching correction along with the control system worked accurately as a whole. It was found that the off-line simulation showed promising and satisfactory results.

5. A state-of-the-art real-time simulator was installed in the RTX-LAB to perform HIL simulation. A brief introduction of the simulator was given. For both the HIL simulation and the experiment, a digital controller was used which implemented the control scheme described earlier. For this purpose, a Target node of the real-time simulator was used. The node has the capability to send and receive both analog and digital signals to and from the external hardware. SIMULINK blocksets along with control tool box and XSG blockset are used to implement the controller. The control algorithms were processed in another Target node which produced reference signals for PWM. The PWM was produced inside the FPGA which has $10ns$ resolution and was sufficient to generate PWM gating signals for the VSC-HVDC system. The reason for using FPGA to generate PWM was to exploit the high resolution capabilities of FPGAs which ultimately produced an accurate PWM.

6. For HIL simulation, the VSC-HVDC system was modeled in C-program and then wrapped up in an S-function in the SIMULINK environment. The model was then loaded in the simulator node of the real-time simulator which was used as the main computation engine for the HIL setup. The system took $0 - 15V$ gate pulses as the input and produced $\pm 15V$ analog feedback signals as the outputs. These outputs were then passed to the controller which was physically connected with the simulator through I/O ports.

7. An experimental setup was installed in the lab which was allowed to transfer a maximum 4kW of active power from one AC side to the other. Even though the converter rating is higher, the power transfer was restricted due to the

limitations of other hardware involved in the experiment. The same controller used for the HIL simulation was used to generate the gate pulses which were fed to the VSCs in the experimental setup.

8. For all three cases: off-line C-program, HIL simulation, and the experiment, closed loop cases were studied where the system went through steady-state and transient situations at various power levels. Close agreement between all these sets of results were observed, especially, HIL simulation showed promising results by being closest to the experimental results. Minor discrepancies are mainly due to the reasons listed in the earlier chapter. The matching of the results corroborates the fact that the developed model, the discretization algorithms, the switching corrections, and the developed control scheme was accurate and efficient.

6.1 Future Work

Importance of HIL simulation in the industry is growing day by day. Accuracy in every step of the simulation is therefore essential for making the simulation reliable and trustworthy. Despite the fact of maximizing the efforts to make the simulation results accurate and reliable, some of the limitations due to insufficient resources and time constraints needs to be addressed in further research in this area. Those limitations are as follows:

- In this research, the voltage source converters are modeled using switching function models instead of modeling individual IGBTs with their V-I characteristics. The modeling of power electronic components with their exact behavior including non-linearity would produce more accurate results. Snubber circuits and di/dt protections are also ignored. Individual component modeling will produce the exact physics for each device which is not possible in a switching function model. In addition, thermal stresses can also be modeled which will help to include detailed protection circuits in the simulation. Undoubtedly, inclusion of a detailed model including all these features will be a challenge for real-time simulation but with the advancement of high speed computing devices along with the adoption of parallel processing technology, this might be a reality very soon.

- High speed FPGAs as core computing engines are gaining foundation for simulating small networks. Development of high performance FPGAs with large number of logic cells might pave the way for this technology to be popular in the simulation industry. Adoption of such FPGAs as the main computing engine will bring down the simulation time-step to nano-seconds which is a fast growing avenue to explore.
- Improvement in the control scheme will provide better immunity to the system when it deals with any uncertainty. In the real-field, systems may suffer from certain degree of unbalance and the control needs to work under those circumstances. Therefore, controllers designed using robust control methods might be able to cope with small differences between the true system and the model used for design.
- Practical networks have non-linearities due to saturation in the magnetic cores used in many equipments. Inclusion of such non-linear devices in the system for real-time simulation is a difficult task as they are solved iteratively. However, with the increase of immense computing power of the new generation multi-core processors, parallel processing technology and other digital technologies, maybe it is time to include them in the model.

Bibliography

[1] N. G. Hingorani, L. Gyugyi, *Understanding FACTS: Concepts and Technology of Flexible AC Transmission Systems*, IEEE Press, 2000.

[2] P. C. S. Krishnayya, R. J. Piwko, T. L. Weaver, M. P. Bahrman and A. E. Hammand, "DC Transmission Terminating at Low Short Circuit Ratio Locations," *IEEE Trans. on Power Delivery*, Vol. PWRD-1, No. 3, pp. 308-317, July 1986.

[3] A. E. Hamand, W. Kuhn, "A Computer Algorithm for Assessing Voltage Stability at AC/DC interconnections" *IEEE Trans. on Power Systems*, Vol. PWRS-1, No. 1, pp. 209-216, February 1986.

[4] P. Kundur, *Power System Stability and Control*, McGraw Hill, 1994

[5] N. G. Hingorani, S. L. Nilson, R. V. Pohl, C. E. Grund, L. Creasap, "Active and Reactive Power Modulation of HVDC Systems" *Proceedings of the IEEE Conference on Over-voltages and Compensation in Integrated AC-DC System* Winipeg, July 1980.

[6] G. Asplund, K. Eriksson, and K. Svensson, "DC Transmission Based on Voltage Source Converters", *Proc. of CIGRE SC14 Colloq.*, Pretoria, South Africa, 1997.

[7] U. Axelsson, A. Holm, C. Liljegren, K. Eriksson, and L.Weimers, "Gotland HVDC light transmissionWorlds first commercial small scale dc transmission", *Proc. of CIRED99 Conf.*, Nice, France, May 1999.

[8] G. Asplund, "Application of HVDC light to power system enhancement", *Proc. of IEEE Winter Power Meeting*, Singapore, Jan. 2000.

[9] V. R. Dinavahi and M. R. Iravani, "Coupling Digital Controllers with Real-Time Digital Simulators for Switched Power Systems", *Proc. Int. Conf. on Power System Transients*, Rio de Janeiro, Brazil, Jun. 2001.

[10] RTDS Technology Inc, "Real-Time Simulation", *Brochure*, Manitoba, Canada.

[11] P. G. McLaren, R. Kuffel, R. Wierckx, J. Giesbrecht, L. Arendt, "A Real-Time Digital Simulator for Testing Relays", *IEEE Trans. on Power Delivery*, Vol. 7, No. 1, pp. 207-213, January 1992.

[12] M. Park, In-Keun Yu, "A novel Real-Time Simulation Technique of Photovoltaic Generation Systems Using RTDS", *IEEE Trans. on Energy Conversion*, Vol. 19, No. 1, pp. 164 - 169, March 2004.

[13] L-F. Pak, M. O. Faruque, X. Nie and V. Dinavahi, "A Versatile Cluster-Based Real-Time Digital Simulator for Power Engineering Research", *IEEE Trans. on Power Systems*, vol. 21, no. 2, pp. 455-465, May 2006.

[14] S. Abourida, C. Dufour, J. Belanger, "Real-time PC-based Simulator of Electric Systems and Drives", *Proc. of 17th IEEE Conference and Exposition on Applied Power Electronics, APEC 2002*, Vol. 1, pp. 433-438, March 10-14, 2002.

[15] C. Dufour, J. Belanger, "Real-Time PC-based Simulator of Electric Systems and Drives", *Proc. of International Conference on Parallel Computing in Electrical Engineering, 2004*, Vol. 1, pp. 105-113, September 7-10 2004.

[16] C. Dufour, J. Belanger, "A Real-Time Simulator for Doubly Fed Induction Generator Based Wind Turbine Applications" *Proc. of IEEE 35th Annual Power Electronics Specialists Conference*, 2004. PESC 04. Vol. 5, pp. 3597 - 3603, 20-25 June 2004.

[17] M. O. Faruque, V. Dinavahi, "An Advanced PC-Cluster Based Real-Time Simulator for Power Electronics and Drives", *Proc. of 2006 IEEE International Symposium on Industrial Electronics*, Vol. 3, pp. 2579 - 2584, July 2006.

[18] A. O. Barry, F. Guay, S. Guerette, P. Giroux, "Digital Real-Time Simulation for Distribution Systems", *Proc. of 2000 IEEE 9th International Conference on Transmission and Distribution Construction, Operation and Live-Line Maintenance*, pp. 252 - 258, 8-12 Oct. 2000.

[19] Lok-Fu Pak, V. Dinavahi, Gary Chang, M. Steurer, P.F. Ribeiro, "Real-Time Digital Time-Varying Harmonic Modeling and Simulation Techniques", IEEE Task Force on Harmonics Modeling and Simulation, *IEEE Trans. on Power Delivery*, Vol. 22, No. 2, pp. 1218 - 1227, April 2007.

[20] G. Sybille, P. Giroux, "Simulation of FACTS Controllers using the MATLAB Power System Blockset and Hypersim Real-Time Simulator", *Proc. of IEEE Power Engineering Society Winter Meeting, 2002*, Vol. 1, pp. 488 - 491, Jan. 27-31, 2002.

[21] A. Monti, E. Santi, R.A. Dougal, M. Riva, "Rapid Prototyping of Digital Controls for Power Electronics", *IEEE Trans. on Power Electronics*, Vol. 18, No. 3, pp. 915 - 923, May 2003.

[22] A. Rubaai, A. R. Ofoli, D. Cobbinah, "DSP-Based Real-Time Implementation of a Hybrid $H\infty$ Adaptive Fuzzy Tracking Controller for Servo-Motor Drives", *IEEE Trans. on Industry Applications*, Vol. 43, No. 2, pp. 476 - 484, March-april 2007.

[23] Mi-Ching Tsai, Chung-Chi Chou, Min-Fu Hsieh, "Development of a Real-Time Servo Control Test Bench", *IEEE Trans. on Education*, Vol. 40, No. 4, pp. 242 - 252, Nov. 1997.

[24] I. Etxeberria-Otadui, V. Manzo, S. Bacha, F. Baltes, "Generalized Average Modelling of FACTS for Real-Time Simulation in ARENE", *IEEE 2002 28th Annual Conference of the Industrial Electronics Society, IECON 02*, Vol. 2, pp. 864 - 869, 5-8 Nov. 2002.

[25] R. Krebs, O. Ruhle, "NETOMAC real-time simulator - a new generation of standard test modules for enhanced relay testing", *Proc. of Eighth IEE*

International Conference on Developments in Power System Protection, 2004, Vol. 2, pp. 669 - 674, 5-8 April 2004.

[26] P. Lehn, J. Rittiger, B. Kulicke, "Comparison of the ATP Version of the EMTP and the NETOMAC Program for Simulation of HVDC Systems", *IEEE Trans. on Power Delivery*, Vol. 10, No. 4, pp. 2048 - 2053, Oct. 1995.

[27] J. M. Javahir, J. Arrillaga, N. R. Watson, "Hybrid Electromagnetic Transient Simulation with the State Variable Representation of HVDC Converter Plant", *IEEE Trans. on Power Delivery*, Vol. 8, No. 3, pp. 1591-1598, July 1993.

[28] M. O. Faruque, V. Dinavahi, W. Xu, "Algorithms for the Accounting of Multiple Switching Events in Digital Simulation of Power Electronic Systems", *IEEE Trans. on Power Delivery*, Vol. 20, No. 2, Part-I, pp. 1157 - 1167, April 2005.

[29] T. Berry, A. R. Daniels, R. W Dunn, "Real-Time Simulation of Power System Transient Behaviour", *Proceedings of 3rd International Conference on Power System Monitoring and Control*, pp.122-127, June 1991.

[30] D. Jakominich, R. Krebs, D. Retzmann, A. Kumar, "Real-Time Digital Power System Simulator Design Considerations and Relay Performance Evaluation", *IEEE Trans. on Power Delivery*, Vol. 14, No. 3, pp. 773-781, July 1999.

[31] M. Kezunovic, V. Skendzic, M. Aganagic, J. K. Bladow, and S. M. McKenna, Design, Implementation and Validation of a Real-Time Digital Simulator for Protective Relay Testing, *IEEE Trans. on Power Delivery*, vol. 11, no. 1, pp. 158164, Jan. 1996.

[32] M. O. Faruque, V. Dinavahi, S. Santoso, R. Adapa, "Review of Electromagnetic Transient Models for non-VSC FACTS", *IEEE Trans. on Power Delivery*, vol. 20, no. 2, pp. 1068-1079, April 2005.

[33] H. W. Dommel, "Digital Computer Solution of Electromagnetic Transients in Single and Multi-Phase Networks", *IEEE Trans. on Power Apparatus and Systems*, Vol. 88, No. 4, pp. 388-395, April 1969.

[34] N. R. Watson and J. Arrillaga, *Power Systems Electromagnetic Transients Simulation*, IEE Press, 2003.

[35] S. Y. R. Hui, C. Christopoulos, "Discrete Transform Technique for Solving Coupled Integro-Differential Equations in Digital Computers", *IEE Proceedings-A*, Vol.138, No. 5, pp. 273-280, September 1991.

[36] A. M. Gole, A. Keri, C. Nwankpa, E. W. Gunther, H. W. Dommel, I. Hassan, J. R. Marti, J. A. Martinez, K. G. Fehrle, L.Tang, M. F. McGranaghan, O. B. Nayak, P. F. Ribeiro, R. Iravani, R. Lasseter, "Guidelines for Modeling Power Electronics in Electric Power Engineering Applications", *IEEE Trans. on Power Delivery*, Vol. 12, No. 1, pp. 505-514, January 1997.

[37] G. G. Parma, V. Dinavahi, "Real-Time Digital Hardware Simulation of Power Electronics and Drives", *IEEE Transactions on Power Delivery*, Vol. 22, No. 2, pp. 1235 - 1246, April 2007.

[38] H. Jin, "Behaviour-Mode Simulation of Power Electronic Circuits", *IEEE Trans. on Power Electronics*, Vol. 12, No. 3, pp. 443-452, May 1997.

[39] S. Cuk and R. D. Middlebrook, "A General Unified Approach to Modeling Switching DC-to-DC Converters in Discontinuous Conduction Mode", *Proc. IEEE PESC77*, pp. 3657, 1977.

[40] V. R. Dinavahi, M. R. Iravani, and R. Bonert, "Real-Time Digital Simulation of Power Electronic Apparatus Interfaced With Digital Controllers", *IEEE Trans. on Power Delivery*, Vol. 16, No. 4, pp. 775-781, October 2001.

[41] A. M. Gole, V. K. Sood, "A Static Compensator Model for Use with Electromagnetic Transient Simulation Programs", *IEEE Trans. on Power Delivery*, Vol. 5, No. 3, pp. 1398-1405, July 1990.

[42] T. L. Maguire, A. M. Gole, "Digital Simulation of Flexible Topology Power Electronic Apparatus in Power Systems", *IEEE Trans. on Power Deivery*, Vol. 6, No. 4, pp. 1831-1840, October 1991.

[43] B. D. Kelper, L. A. Dessaint, K. A Haddad, H. Nakra, "A Comprehensive Approach to Fixed-Step Simulation of Switched Circuits ", *IEEE Trans. on Power Electronics,* Vol. 17, No. 2, pp. 216-224, March 2002.

[44] G. D. Irwin, D. A. Woodford, A. Gole, "Precision Simulation of PWM controller", *Proceedings of International Conference on Power System Transients (IPST)* 2001.

[45] V. Q. Do, D. McCallum, P. Giroux, B. D. Kelper, "A Backward-Forward Interpolation Technique for a Precise Modeling of Power Electronics in HYPERSIM", *IPST'01-International Conference on Power Systems Transients, Rio de Janeiro, Brazil,* pp. 337-342, June 2001.

[46] K. Strunz, L. R. Linares, J. R. Marti, O. Huet, X. Lombard, "Efficient and Accurate Representation of Asynchronous Network Structure Changing Phenomena in Digital Real-Time Simulators", *IEEE Trans. on Power Systems,* Vol. 15, No. 2, pp. 586-592, May 2000.

[47] Manitoba HVDC Research Centre, *PSCAD/EMTDC Brochure,* 2001.

[48] M. Zou, J. Mahseredjian, G. Joos, B. Delourme, L. G. Lajoie, "Interpolation and Reinitialization for the Simulation of Power Electronic Circuits", *Proceedings of International Conference on Power System Transients-IPST'03,* Sept 28-October 2, 2003.

[49] R. Kuffel, J. Giesbrecht, T. Maguire, R. P.Wierckx, and P. G. McLaren, "RTDS-A Fully Digital Power System Simulator Operating in Real-Time," *Proc. EMPD,* vol. 2, pp. 498503, 1995.

[50] C. Schauder, H. Mehta, "Vector Analysis and Control of Advanced Static VAR Compensators", *IEE Proceedings-C,* Vol. 140, No. 4, pp. 299-306, July 1993.

[51] J. Holtz, "Pulsewidth Modulation -A Aurvey", *IEEE Transactions on Industrial Electronics,* Vol. 39, No. 5, Dec. 1992, pp. 410-419.

[52] X. Wang, R. M. Mathur, "Real-Time Digital Simulator of the Electromagnetic Transients of Transmission Lines with Frequency Dependence", *IEEE Trans. on Power Delivery*, Vol. 4, No. 4, pp. 2249-2255, October 1989.

[53] P. J. Throckmorton, L. Wozniak, "A Generic DSP-Based Real-Time Simulator with Application to Hydrogenerator Speed Controller Development", *IEEE Trans. on Energy Conversion*, Vol. 9, No. 2, pp. 238-242, June 1994.

[54] X. Wang, D. A. Woodford, R. Kuffel, R. Wierckx, "A Real-Time Transmission Line Model for a Digital TNA", *IEEE Trans. on Power Delivery*, Vol. 11, No. 2, pp. 1092-1097, April 1996.

[55] C. Dufour, Le-Huy Hoang, J.C. Soumagne, A. El Hakimi, "Real-Time Simulation of Power Transmission Lines using Marti Model with Optimal Fitting on Dual-DSP Card", *IEEE Trans. on Power Delivery*, Vol. 11, No. 1, pp. 412-419, January 1996.

[56] V. Dinavahi, M. R. Iravani, R. Bonert, "Design of a Real-Time Digital Simulator for a D-STATCOM System", *IEEE Trans. on Industrial Electronics*, Vol. 51, No. 5, pp. 1001-1008, October 2004.

[57] Y. Li, D. M. Vilathgamuwa, P. C. Loh, "Design, Analysis, and Real-Time Testing of a Controller for Multibus Microgrid System", *IEEE Trans. on Power Electronics*, Vol. 19, No. 5, pp. 1195-1204, September 2004.

[58] M. Foley, Y. Chen, A. Bose, "A Real-Time Power System Simulation Laboratory Environment", *IEEE Trans. on Power Systems*, Vol. 5, No. 4, pp. 1400-1406, November 1990.

[59] J. R. Marti, L. R. Linares, "Real-Time EMTP-Based Transients Simulation", *IEEE Trans. on Power Systems*, Vol. 9, No. 3, pp. 1309-1317, August 1994.

[60] O. Devaux, L. Lavacher, O. Huet, "An Advanced and Powerful Real-Time Digital Transient Network Analyser", *IEEE Trans. on Power Delivery*, Vol. 13, No. 2, pp. 421-426, April 1998.

[61] H. Taoka, I. Iyoda, H. Noguchi, N. Sato, T. Nakazawa, "Real-Time Digital Simulator for Power System Analysis on a Hypercube Computer", *IEEE Trans. on Power Systems*, Vol. 7, No. 1, pp. 1-10, February 1992.

[62] J. A. Hollman, J. R. Marti, "Real-Time Network Simulation with PC-cluster", *IEEE Trans. on Power Systems*, Vol. 18, No. 2, pp. 563-569, May 2003.

[63] I. Etxeberria-Otadui, V. Manzo, S. Bacha, F. Baltes, "Generalized Average Modelling of FACTS for Real-Time Simulation in ARENE", *Proceedings of IEEE 2002 28th Annual Conference, IECON 02*, Vol. 2, pp. 864 - 869, Nov. 2002.

[64] R. Krebs, O. Ruhle, "NETOMAC Real-Time Simulator - A New Generation of Standard Test Modules for Enhanced Relay Testing", Proceedings of Eighth IEE International Conference on Developments in Power System Protection, Vol. 2, pp. 669 - 674, April 2004.

[65] N. K. Sinha and B. Kuszta, *Modeling and Identification of Dynamic Systems*, Van Nostrand Reinhold Company Inc., 1983.

[66] L. O. Chua and P. M. Lin, *Computer Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*, Prentice-Hall, INC., 1975.

[67] Chi-Tsong Chen, *Analog and Digital Control System Design: Transfer-Function, State-Space, and Algebraic Methods*, Saunders College Publishing, 1993.

[68] N. R. Watson and G. D. Irwin, "Comparison of Root-Matching Techniques for Electromagnetic Transient Simulation", *IEE Trans. on Power Delivery*, vol. 15, no. 2, April 2000.

[69] J. R. Marti and J. lin, "Suppression of Numerical Oscillation in the EMTP", *IEEE Trans. on Power System*, vol. 4, no. 2, May 1989.

[70] Cuiquing Du, *The Control of VSC HVDC and Its Use for Large Industrial Power Systems*, Ph.D thesis, Chalmers University of Technology, 2003.

[71] Power System Blockset Users Guide, *TEQSIM International, Inc.*, 2001.

[72] T. Chen and B. Francis, *Optimal Sampled Data Control System*, Springer, 1995.

[73] M. O. Faruque, Y. Zhang, V. Dinavahi, "Detailed Modeling of CIGRE HVDC Benchmark System Using PSCAD/EMTDC and PSB/SIMULINK", *IEEE Trans. on Power Delivery*, Vol. 21, No. 1, Jan. 2006, pp. 378-387.

[74] R. Pena, J. C. Clare, G. M. Asher, "Doubly Fed Induction Generator using Back-to-Back PWM Converters and Its Application to Variable Speed Wind-Energy Generation", *IEE Proc. Electrical Power Appl.*, Vol. 143, No. 3, pp. 231-241, May 1996.

Appendix A

A.1 SIT and RIT for R, L, C, RL, RC, LC and RLC Network

In this section, the time-domain difference equations for R, L, C, RL, LC, RC and RLC networks are derived using the step-invariant and ramp invariant techniques described in Chapter 2.

A.1.1 Resistive Network

Fig. A.1(a) shows a resistive network where a resistor, R is connected in series with a voltage source, $v(t)$, produces a current $i(t)$. The linear relationship between $v(t)$ and $i(t)$ is given as;

$$v(t) = Ri(t) \qquad \Rightarrow \qquad \frac{i(t)}{v(t)} = \frac{1}{R} \tag{A.1}$$

Now taking the Laplace transform of (A.1),

$$G(s) = \frac{I(s)}{V(s)} = \frac{1}{R} \tag{A.2}$$

Using the step-invariant transformation (SIT) as described in Chapter 2,

$$\begin{aligned} G(z) = \frac{I(z)}{V(z)} &= \frac{z-1}{z}\mathcal{Z}\left[\frac{G(s)}{s}\right] \Rightarrow \frac{z-1}{z}\mathcal{Z}\left[\frac{1}{Rs}\right] \Rightarrow \frac{z-1}{z} \times \frac{1}{R}\frac{z}{z-1} \\ &= \frac{1}{R} \end{aligned} \tag{A.3}$$

Converting (A.3) back to time domain, we have

$$i(k\Delta t) = \frac{1}{R}v(k\Delta t) \tag{A.4}$$

For Ramp invariant transformation,

$$\begin{aligned} G(z) = \frac{I(z)}{V(z)} &= \frac{(z-1)^2}{\Delta tz}\mathcal{Z}\left[\frac{G(s)}{s^2}\right] \Rightarrow \frac{(z-1)^2}{\Delta tz}\mathcal{Z}\left[\frac{1}{Rs^2}\right] \\ &\Rightarrow \frac{(z-1)^2}{\Delta tz} \times \frac{1}{R}\frac{\Delta tz}{(z-1)^2} = \frac{1}{R} \end{aligned} \tag{A.5}$$

Time domain conversion of (A.5) yields;

$$i(k\Delta t) = \frac{1}{R}v(k\Delta t) \tag{A.6}$$

A.1.2 Inductive Network

The relationship between the current $i(t)$ and voltage $v(t)$ in the inductive circuit in Fig. A.1(b) can be as:

$$L\frac{di(t)}{dt} = v(t) \tag{A.7}$$

Taking the Laplace transform and assuming that $i(0^-) = 0$ we get,

$$G(s) = \frac{I(s)}{V(s)} = \frac{1}{Ls} \tag{A.8}$$

Using the step-invariant transformation (SIT),

$$\begin{aligned} G(z) = \frac{I(z)}{V(z)} &= \frac{z-1}{z}\mathcal{Z}\left[\frac{G(s)}{s}\right] \Rightarrow \frac{z-1}{z}\mathcal{Z}\left[\frac{1}{Ls^2}\right] \\ &\Rightarrow \frac{z-1}{z} \times \frac{1}{L}\frac{\Delta tz}{(z-1)^2} = \frac{\Delta t}{L}\frac{1}{z-1} \end{aligned} \tag{A.9}$$

Rewriting the above equation and then converting it into time domain,

$$\begin{aligned} I(z) &= \frac{\Delta t}{L}z^{-1}V(z) + z^{-1}I(z) \\ i(k\Delta t) &= \frac{\Delta t}{L}v[(k-1)\Delta t] + i[(k-1)\Delta t] \end{aligned} \tag{A.10}$$

Using the Time Shifted Step-Invariant Transformation (TSSIT), the current can be given as,

$$\begin{aligned} I(z) &= \frac{\Delta t}{L}V(z) + z^{-1}I(z) \\ i(k\Delta t) &= \frac{\Delta t}{L}v(k\Delta t) + i[(k-1)\Delta t] \end{aligned} \tag{A.11}$$

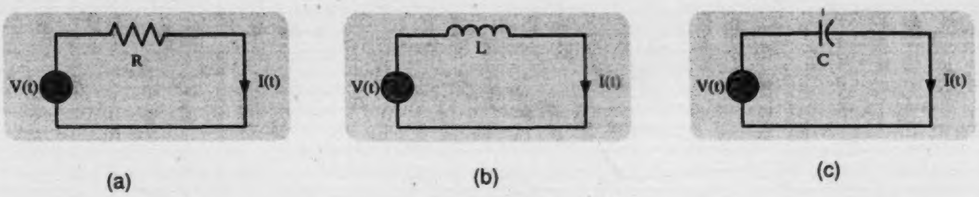


Figure A.1: (a) Resistive network (b) Inductive Network (c) Capacitive network

Using ramp invariant transformation,

$$G(z) = \frac{I(z)}{V(z)} = \frac{(z-1)^2}{\Delta tz}\mathcal{Z}\left[\frac{G(s)}{s^2}\right] \Rightarrow \frac{(z-1)^2}{\Delta tz}\mathcal{Z}\left[\frac{1}{Ls^3}\right]$$

$$\Rightarrow \frac{(z-1)^2}{\Delta tz} \times \frac{1}{2L}\frac{\Delta t^2 z(z+1)}{(z-1)^3} = \frac{\Delta t}{2L}\frac{z+1}{z-1} \qquad (A.12)$$

which gives the time domain equation as,

$$i(k\Delta t) = \frac{\Delta t}{2L}\left[v(k\Delta t) + v[(k-1)]\Delta t\right] + i[(k-1)\Delta t] \qquad (A.13)$$

A.1.3 Capacitive Network

The transfer function of a capacitive network does not have any pole for a voltage source as an input. Hence its discretization is not possible using SIT. However, for a current source as an input, the following equation can be used to get the voltage across the capacitor,

$$v(k\Delta t) = \frac{\Delta t}{C}i[(k-1)\Delta t] + v[(k-1)\Delta t] \qquad (A.14)$$

Its discretization can be evaluated using other techniques or it can be done when combined with R or L elements.

A.1.4 RL Network

For RL network as shown in Fig. A.2(a), the linear dynamics can be expressed as;

$$v(t) = L\frac{di(t)}{dt} + Ri(t) \qquad (A.15)$$

Laplace transform of the above equation gives;

$$G(s) = \frac{I(s)}{V(s)} = \frac{1}{R+sL} \qquad (A.16)$$

Using SIT, the z-domain transfer function of (A.16) can be found as;

$$\begin{aligned}
G(z) &= \frac{I(z)}{V(z)} = \frac{z-1}{z}\mathcal{Z}\left[\frac{G(s)}{s}\right] \quad \Rightarrow \frac{z-1}{z}\mathcal{Z}\left[\frac{1}{s(R+sL)}\right] \\
&\Rightarrow \frac{z-1}{z}\mathcal{Z}\left[\frac{1}{Rs} - \frac{1}{R(s+R/L)}\right] \\
\frac{I(z)}{V(z)} &= \frac{z-1}{z}\left[\frac{1}{R}\frac{z}{z-1} - \frac{1}{R}\frac{z}{z-e^{-\frac{R\Delta t}{L}}}\right] = \frac{1}{R}\left[\frac{1-e^{-\frac{R\Delta t}{L}}}{z-e^{-\frac{R\Delta t}{L}}}\right] \\
&= \frac{1}{R}\left[\frac{(1-e^{-\frac{R\Delta t}{L}})z^{-1}}{(1-e^{-\frac{R\Delta t}{L}}z^{-1})}\right] \\
I(z) &= \frac{1}{R}\left(1-e^{-\frac{R\Delta t}{L}}\right)z^{-1}V(z) + e^{-\frac{R\Delta t}{L}}z^{-1}I(z) \qquad \text{(A.17)}
\end{aligned}$$

Converting (A.17) into time domain,

$$i(k\Delta t) = \frac{1}{R}\left(1-e^{-\frac{R\Delta t}{L}}\right)v[(k-1)\Delta t] + e^{-\frac{R\Delta t}{L}}i[(k-1)\Delta t] \qquad \text{(A.18)}$$

This is an explicit algorithm and using the compensated step-invariant transformation (TSSIT), the algorithm becomes implicit.

$$i(k\Delta t) = \frac{1}{R}\left(1-e^{-\frac{R\Delta t}{L}}\right)v(k\Delta t) + e^{-\frac{R\Delta t}{L}}i[(k-1)\Delta t] \qquad \text{(A.19)}$$

Using ramp-invariant transformation,

$$\begin{aligned}
G(z) &= \frac{I(z)}{V(z)} = \frac{(z-1)^2}{\Delta t z}\mathcal{Z}\left[\frac{G(s)}{s^2}\right] \quad \Rightarrow \frac{(z-1)^2}{\Delta t z}\mathcal{Z}\left[\frac{1}{s^2(R+sL)}\right] \\
&= \frac{(z-1)^2}{\Delta t z}\mathcal{Z}\left[-\frac{L}{R^2}\frac{1}{s} + \frac{1}{R}\frac{1}{s^2} + \frac{L}{R^2}\frac{1}{(s+R/L)}\right] \\
\frac{I(z)}{V(z)} &= \frac{(z-1)^2}{\Delta t z}\left[-\frac{L}{R^2}\frac{z}{z-1} + \frac{1}{R}\frac{\Delta t z}{(z-1)^2} + \frac{L}{R^2}\frac{z}{(z-e^{-\frac{R\Delta t}{L}})}\right] \\
&= \frac{\left(R\Delta t - L + Le^{-\frac{R\Delta t}{L}}\right) + \left(L - Le^{-\frac{R\Delta t}{L}} - R\Delta t e^{-\frac{R\Delta t}{L}}\right)z^{-1}}{R^2\Delta t\left(1-e^{-\frac{R\Delta t}{L}}z^{-1}\right)}
\end{aligned}$$

Converting it into time domain,

$$\begin{aligned}
i(k\Delta t) &= \frac{\left(R\Delta t - L + Le^{-\frac{R\Delta t}{L}}\right)}{R^2\Delta t}v(k\Delta t) \\
&+ \frac{\left(L - Le^{-\frac{R\Delta t}{L}} - R\Delta t e^{-\frac{R\Delta t}{L}}\right)}{R^2\Delta t}v[(k-1)\Delta t] + e^{-\frac{R\Delta t}{L}}i[(k-1)\Delta t]
\end{aligned}$$

A.1.5 LC Network

For an LC network shown in Fig. A.2(b), the network equation can be written as;

$$v(t) = L\frac{di(t)}{dt} + \frac{1}{C}\int i(t)dt \tag{A.20}$$

The Laplace Transform of the above equation is

$$\begin{aligned} G(s) = \frac{I(s)}{V(s)} &= \frac{Cs}{LCs^2+1} \quad \Rightarrow \frac{G(s)}{s} = \frac{1}{L}\,\frac{1}{s^2+\frac{1}{LC}} \\ &= \sqrt{\frac{C}{L}}\,\frac{\sqrt{\frac{1}{LC}}}{s^2+\frac{1}{LC}} = \gamma\,\frac{\omega}{s^2+\omega^2} \end{aligned} \tag{A.21}$$

where, $\gamma = \sqrt{\frac{C}{L}}$ and $\omega = \sqrt{\frac{1}{LC}}$. Using SIT, we find,

$$\begin{aligned} G(z) &= \frac{I(z)}{V(z)} = \frac{z-1}{z}\mathcal{Z}\left[\frac{G(s)}{s}\right] \quad \Rightarrow \frac{z-1}{z}\mathcal{Z}\left[\gamma\,\frac{\omega}{s^2+\omega^2}\right] \\ &= \frac{z-1}{z}\,\gamma\,\frac{z\sin\omega\Delta t}{z^2-2z\cos\omega\Delta t+1} \\ \frac{I(z)}{V(z)} &= \frac{\gamma(z-1)\sin\omega\Delta t}{z^2-2z\cos\omega\Delta t+1} = \frac{\gamma\sin\omega\Delta t(z-1)}{z^2-2z\cos\omega\Delta t+1} \\ &= \frac{\gamma\sin\omega\Delta t(z^{-1}-z^{-2})}{1-2z^{-1}\cos\omega\Delta t+z^{-2}} \end{aligned} \tag{A.22}$$

Converting the above equation into time-domain,

$$\begin{aligned} i(k\Delta t) &= \gamma\sin\omega\Delta t v[(k-1)\Delta t] - \gamma\sin\omega\Delta t v[(k-2)\Delta t] \\ &+ 2\cos\omega\Delta t i[(k-1)\Delta t] - i[(k-2)\Delta t] \end{aligned} \tag{A.23}$$

If TSSIT is used, then the expression becomes;

$$\begin{aligned} i(k\Delta t) &= \gamma\sin\omega\Delta t v(k\Delta t) - \gamma\sin\omega\Delta t v[(k-1)\Delta t] \\ &+ 2\cos\omega\Delta t i[(k-1)\Delta t] - i[(k-2)\Delta t] \end{aligned} \tag{A.24}$$

A.1.6 RC Network

For RC network shown in Fig. A.2(c), the network equation can be written as;

$$v(t) = Ri(t) + \frac{1}{C}\int i(t)dt \tag{A.25}$$

whose Laplace Transform can be obtained as;

$$G(s) = \frac{I(s)}{V(s)} = \frac{s}{R(s + \frac{1}{RC})} \tag{A.26}$$

Z-domain transfer function of (A.26) through SIT,

$$\begin{aligned}
G(z) &= \frac{I(z)}{V(z)} = \frac{z-1}{z}\mathcal{Z}\left[\frac{G(s)}{s}\right] \Rightarrow \frac{z-1}{z}\mathcal{Z}\left[\frac{1}{R(s+\frac{1}{RC})}\right] \\
&\Rightarrow \frac{z-1}{z} \times \frac{1}{R} \times \frac{z}{z - e^{-\frac{\Delta t}{RC}}} \\
\frac{I(z)}{V(z)} &= \frac{z-1}{R}\,\frac{1}{(z - e^{-\frac{\Delta t}{RC}})} = \frac{1}{R}\,\frac{(1 - z^{-1})}{(1 - e^{-\frac{\Delta t}{RC}} z^{-1})} \\
I(z) &= \frac{1}{R}\left[1 - z^{-1}\right] V(z) + e^{-\frac{\Delta t}{RC}} z^{-1} I(z)
\end{aligned} \tag{A.27}$$

Conversion into time-domain of equation (A.27) yields the following:

$$i(k\Delta t) = \frac{1}{R}[v(k\Delta t) - v(k-1)\Delta t] + e^{-\frac{\Delta t}{RC}} i[(k-1)\Delta t] \tag{A.28}$$

Using ramp-invariant transformation,

$$\begin{aligned}
G(z) &= \frac{I(z)}{V(z)} = \frac{(z-1)^2}{\Delta t z}\mathcal{Z}\left[\frac{G(s)}{s^2}\right] \Rightarrow \frac{(z-1)^2}{\Delta t z}\mathcal{Z}\left[\frac{1}{Rs(s+\frac{1}{RC})}\right] \\
&= \frac{(z-1)^2}{\Delta t z}\mathcal{Z}\left[\frac{C}{s} - \frac{C}{s + \frac{1}{RC}}\right] \\
\frac{I(z)}{V(z)} &= \frac{(z-1)^2}{\Delta t z}\left[\frac{Cz}{z-1} - \frac{Cz}{z - e^{-\frac{\Delta t}{RC}}}\right] \\
&= \frac{C}{\Delta t}\,\frac{(1 - z^{-1})(1 - e^{-\frac{\Delta t}{RC}})}{(1 - e^{-\frac{\Delta t}{RC}} z^{-1})}
\end{aligned} \tag{A.29}$$

Conversion of the above equation into time-domain gives,

$$i(k\Delta t) = \frac{C}{\Delta t}(1 - e^{-\frac{\Delta t}{RC}})[v(k\Delta t) - v(k-1)\Delta t] + e^{-\frac{\Delta t}{RC}} i[(k-1)\Delta t] \tag{A.30}$$

A.1.7 RLC Network

For the RLC network of Fig. A.2(d), the network equation is

$$v(t) = Ri(t) + \frac{1}{C}\int i(t)dt + L\frac{di(t)}{dt} \tag{A.31}$$

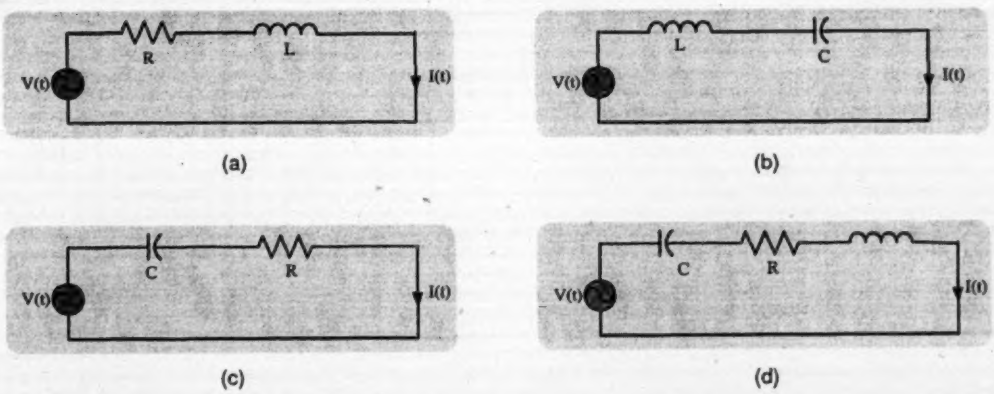


Figure A.2: (a) RL network (b) LC Network (c) CR network (d) RLC network

Taking Laplace Transform and assuming zero initial states,

$$\begin{aligned} G(s) &= \frac{I(s)}{V(s)} = \frac{1}{R + Ls + \frac{1}{CS}} = \frac{Cs}{LCs^2 + RCs + 1} \\ \frac{G(s)}{s} &= \frac{1}{L}\frac{1}{s^2 + \frac{R}{L}s + \frac{1}{LC}} = \frac{1}{L(\beta - \alpha)}\left[\frac{1}{s+\alpha} - \frac{1}{s+\beta}\right] \end{aligned} \tag{A.32}$$

where $\alpha = \frac{-\frac{R}{L} + \sqrt{(\frac{R}{L})^2 - \frac{4}{LC}}}{2}$ and $\beta = \frac{-\frac{R}{L} - \sqrt{(\frac{R}{L})^2 - \frac{4}{LC}}}{2}$ and both of them must be negative. Using SIT, the z-transform can be obtained and given as;

$$\begin{aligned} G(z) &= \frac{I(z)}{V(z)} = \frac{z-1}{z}\mathcal{Z}\left[\frac{G(s)}{s}\right] \\ &= \frac{z-1}{z}\mathcal{Z}\left[\frac{1}{L(\beta - \alpha)}\left(\frac{1}{s+\alpha} - \frac{1}{s+\beta}\right)\right] \\ &= \frac{z-1}{z}\frac{1}{L(\beta-\alpha)}\left[\frac{z}{z - e^{-\alpha\Delta t}} - \frac{z}{z - e^{-\beta\Delta t}}\right] \\ &= \frac{z-1}{L(\beta-\alpha)}\left[\frac{e^{-\alpha\Delta t} - e^{-\beta\Delta t}}{z^2 - (e^{-\alpha\Delta t} + e^{-\beta\Delta t})z + e^{-(\alpha+\beta)\Delta t}}\right] \\ &= \frac{z^{-1} - z^{-2}}{L(\beta-\alpha)}\left[\frac{e^{-\alpha\Delta t} - e^{-\beta\Delta t}}{1 - (e^{-\alpha\Delta t} + e^{-\beta\Delta t})z^{-1} + e^{-(\alpha+\beta)\Delta t}z^{-2}}\right] \end{aligned} \tag{A.33}$$

converting the above equation into time-domain,

$$\begin{aligned} i(k\Delta t) &= \frac{e^{-\alpha\Delta t} - e^{-\beta\Delta t}}{L(\beta - \alpha)}[v(k-1)\Delta t - v(k-2)\Delta t] \\ &+ (e^{-\alpha\Delta t} + e^{-\beta\Delta t})i(k-1)\Delta t - e^{-(\alpha+\beta)\Delta t}i(k-2)\Delta t \end{aligned} \tag{A.34}$$

Using TSSIT, the above equation can be found as;

$$\begin{aligned} i(k\triangle t) &= \frac{e^{-\alpha\triangle t} - e^{-\beta\triangle t}}{L(\beta-\alpha)}\left[v(k\triangle t) - v(k-1)\triangle t\right] \\ &+ (e^{-\alpha\triangle t} + e^{-\beta\triangle t})i(k-1)\triangle t - e^{-(\alpha+\beta)\triangle t}i(k-2)\triangle t \qquad \text{(A.35)} \end{aligned}$$

Appendix B

B.1 Simulation Diagrams of VSC-HVDC

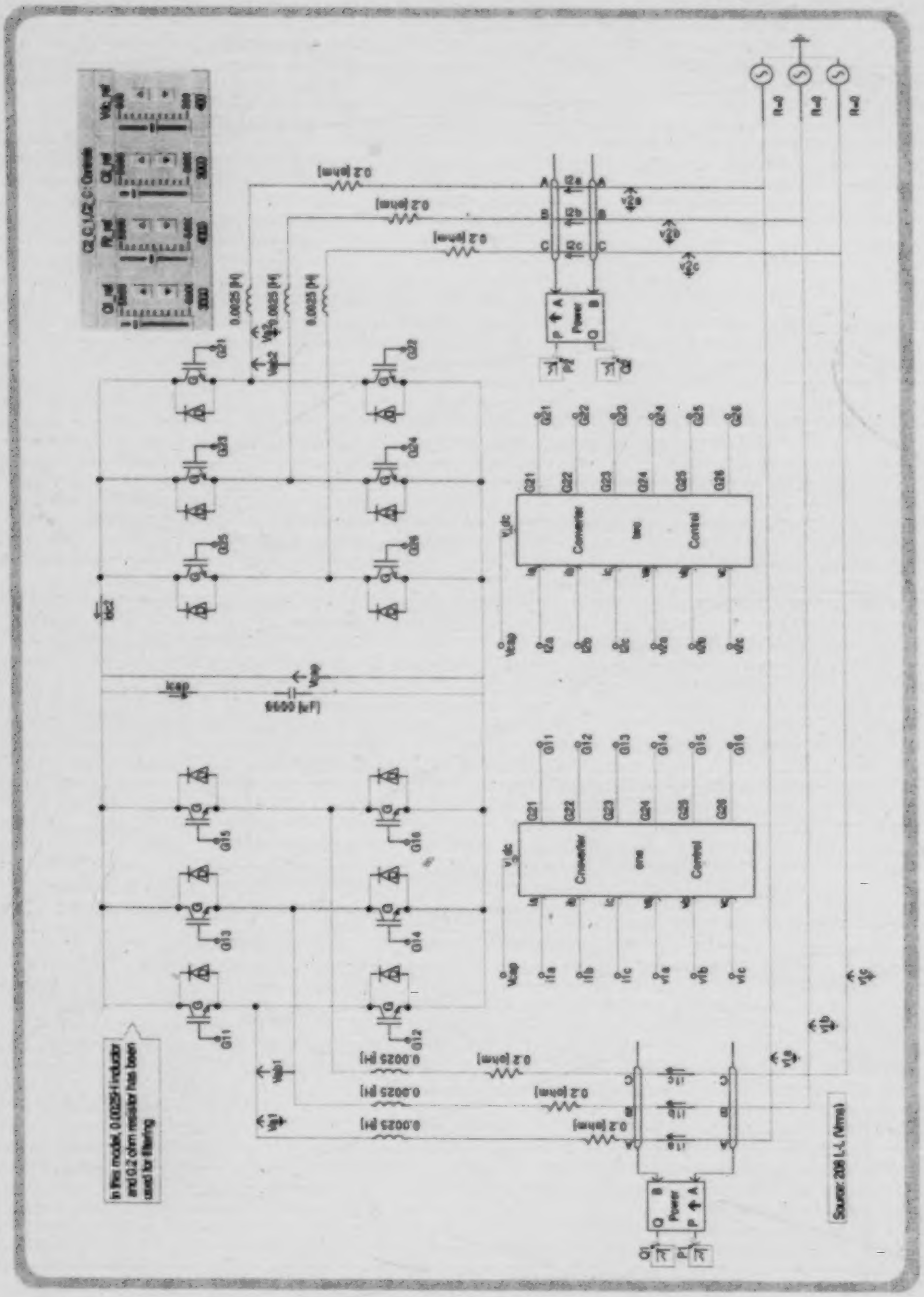


Figure B.1: Simulation of VSC-HVDC using PSCAD/EMTDC

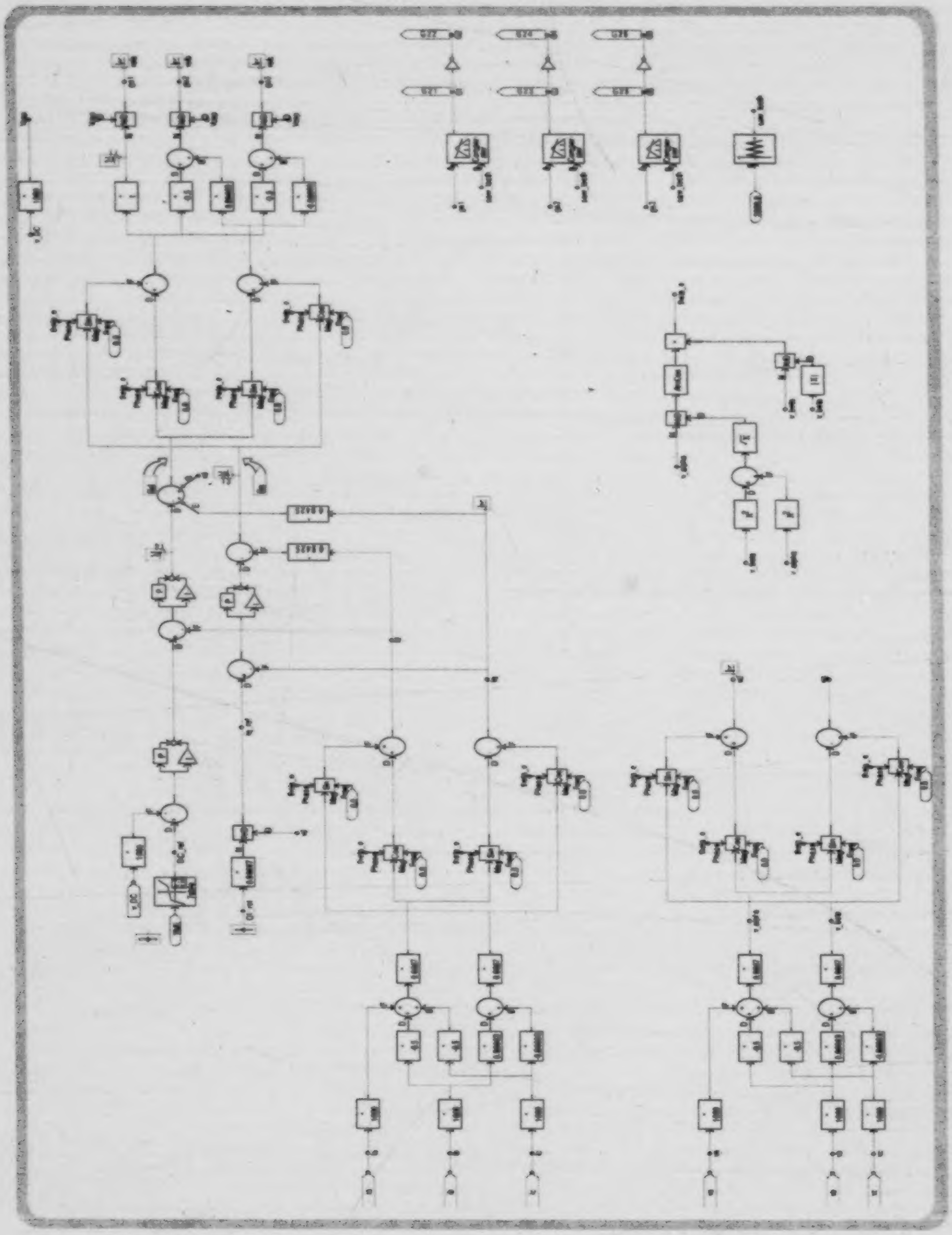


Figure B.2: Controller 1 of VSC-HVDC in PSCAD/EMTDC

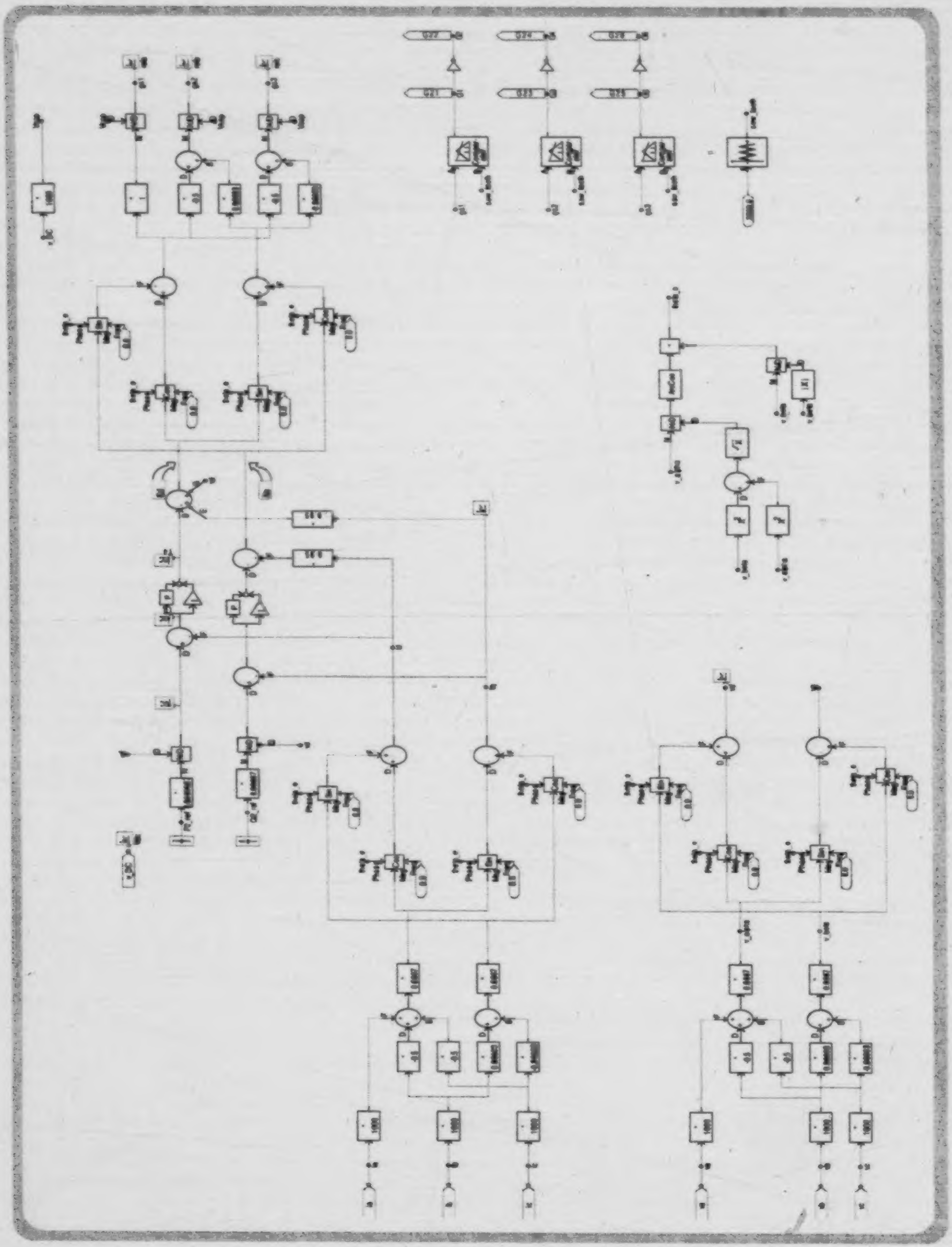


Figure B.3: Controller 2 of VSC-HVDC in PSCAD/EMTDC

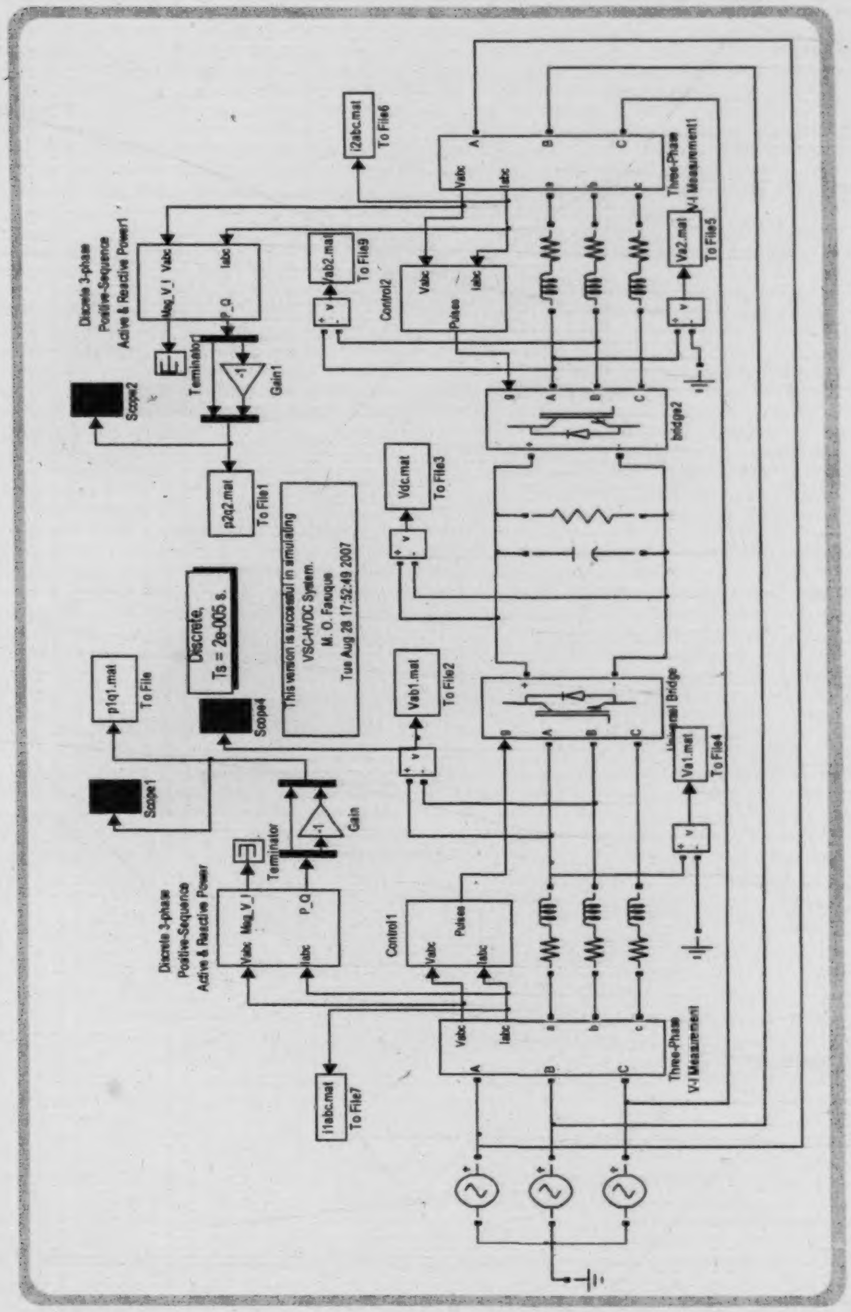
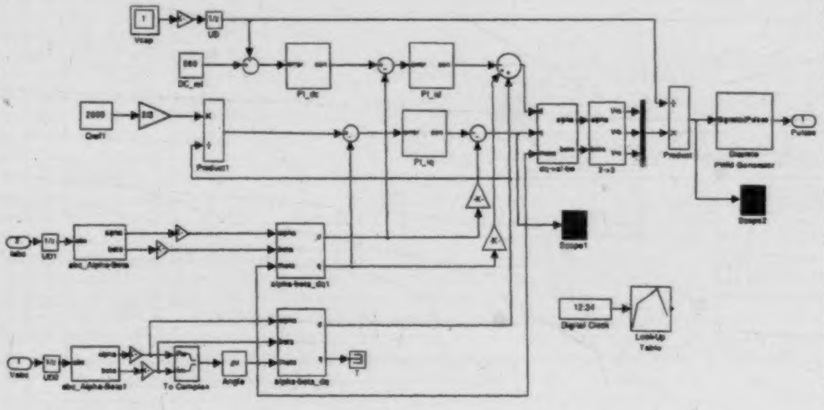
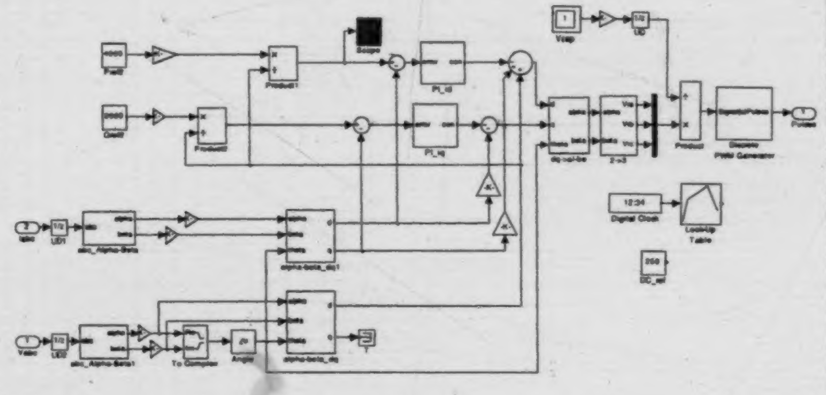


Figure B.4: Simulation of VSC-HVDC using PSB/SIMULINK



(a) Controller for converter 1



(a) Controller for converter 2

Figure B.5: Controllers of VSC-HVDC in PSB/SIMULINK

Appendix C

C.1 Control of VSC-HVDC using SIMULINK and RTLAB Toolbox

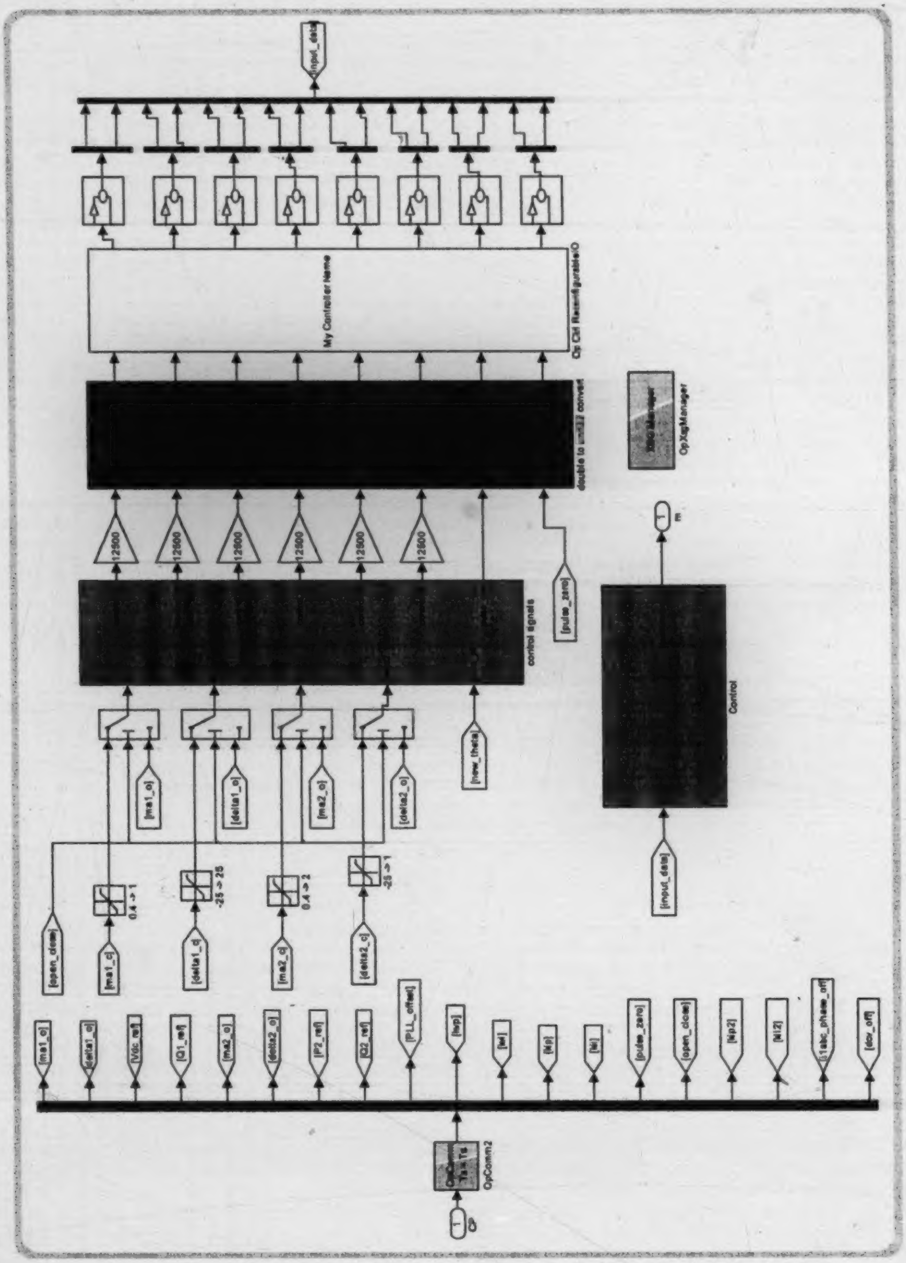


Figure C.1: SIMULINK diagram for the implementation of the controllers for both HIL simulation and experimental validation of VSC-HVDC system.

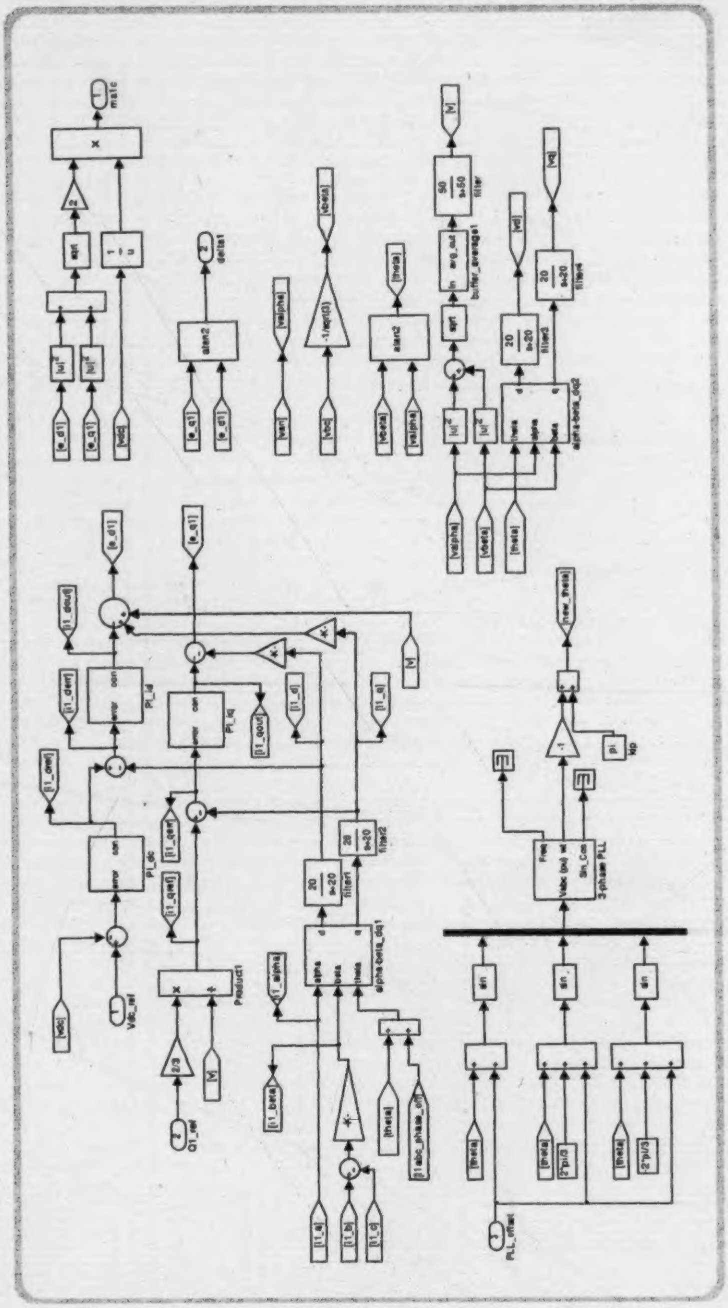


Figure C.2: SIMULINK diagram of Controller 1.

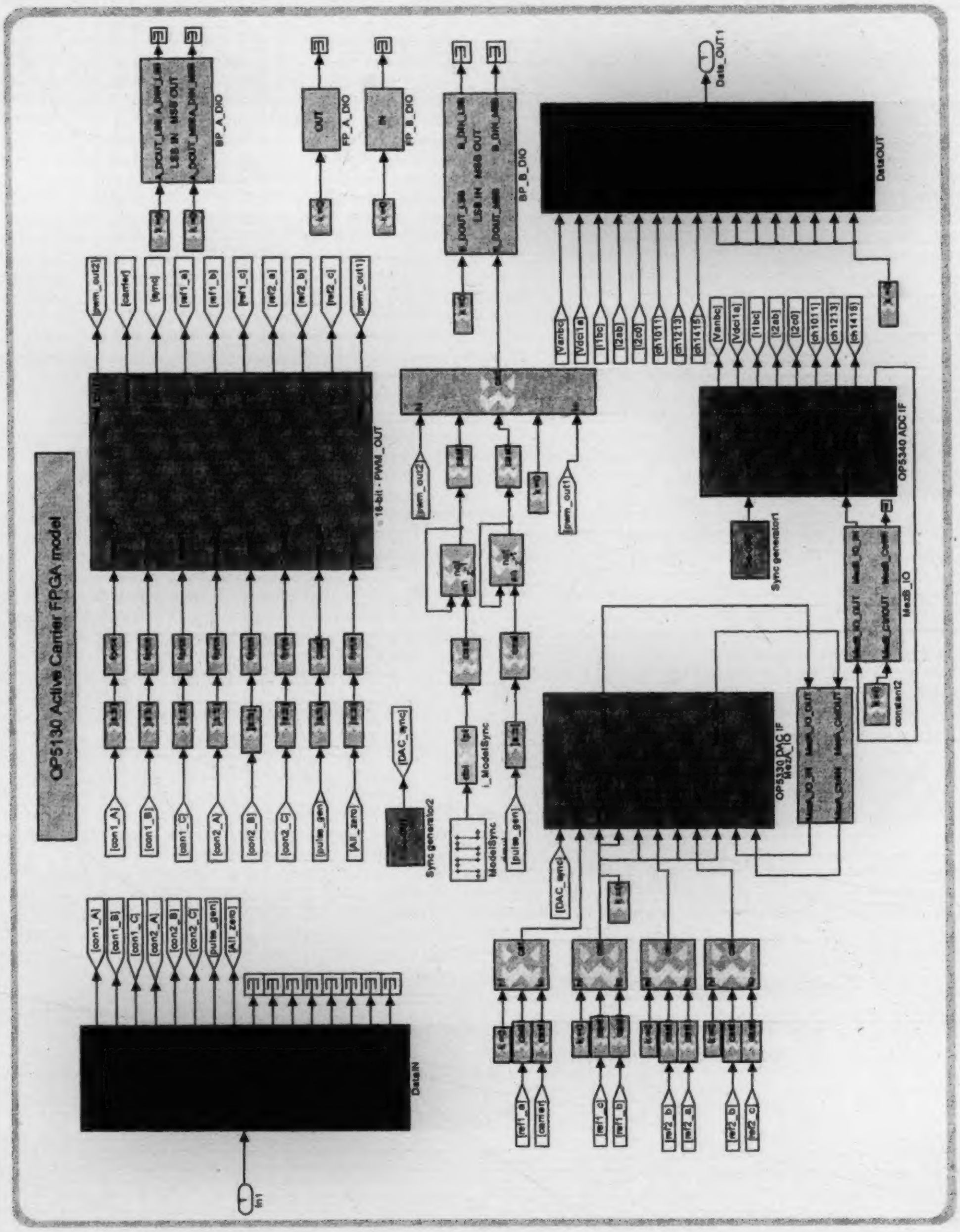


Figure C.3: Generation of PWM using XSG block and FPGA carrier board.

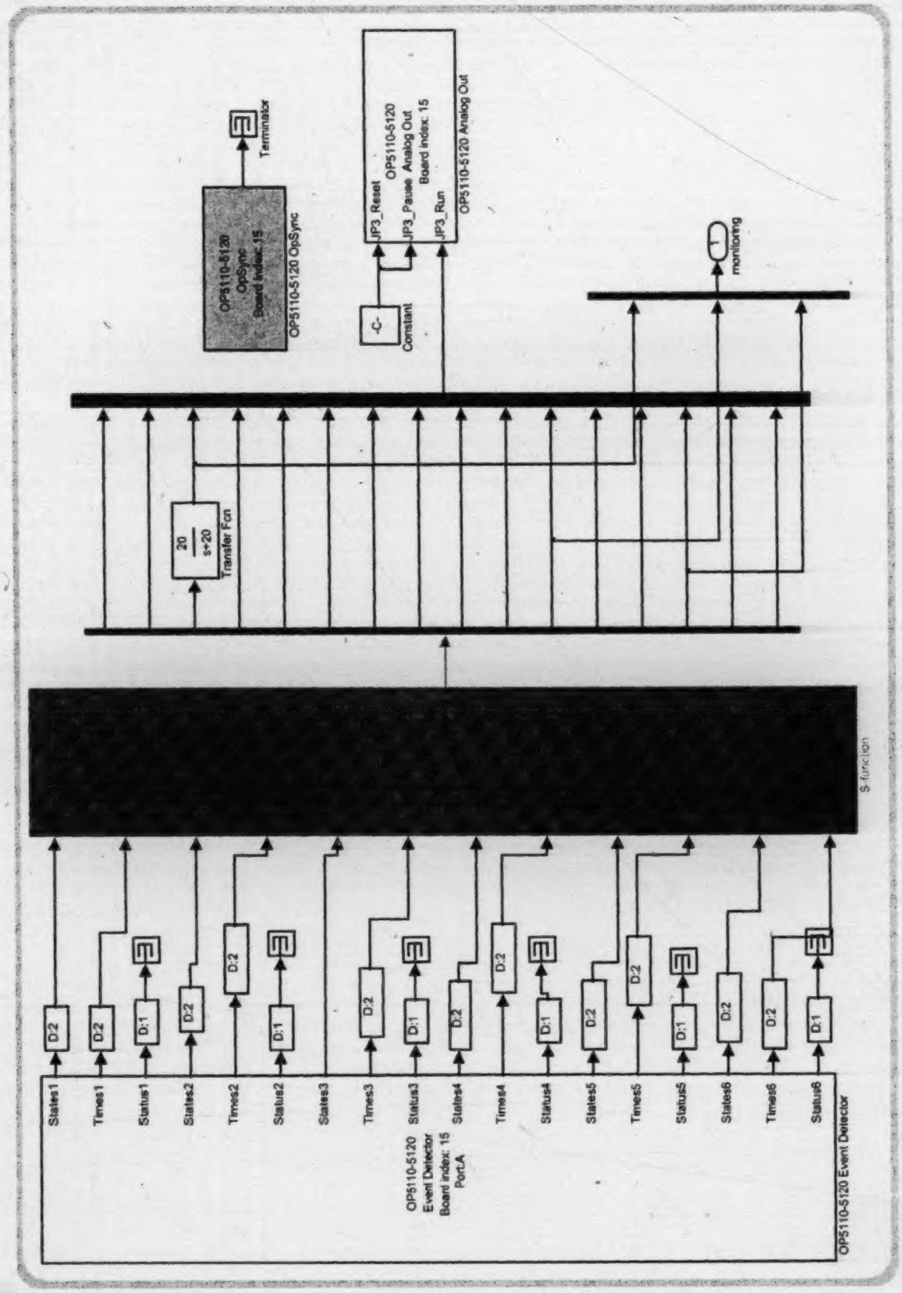


Figure C.4: Simulink and RTLab blocks for HIL-Simulation of VSC-HVDC.